

Математика
программных
систем

Межвузовский сборник научных трудов

Выпуск 15

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Пермский государственный национальный
исследовательский университет»

Федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский университет
"Высшая школа экономики"»
(Пермский филиал)

Государственное образовательное учреждение высшего образования
«Кубанский государственный университет»

МАТЕМАТИКА ПРОГРАММНЫХ СИСТЕМ

Межвузовский сборник научных трудов

Выпуск 15

Пермь 2018

УДК 004.4+004.8+004.9

ББК 32.973

М34

М34 Математика программных систем: межвуз. сб. науч. тр. / под ред. А.И. Микова и Л.Н. Лядовой; Перм. гос. нац. исслед. ун-т. – Пермь, 2018. – Вып. 15. – 112 с.: ил.

ISBN 978-5-7944-3237-4 (вып. 15)

ISBN 978-5-7944-1741-8

В сборник включены статьи, относящиеся к следующим направлениям научной работы: моделирование и анализ процессов и систем; разработка, анализ и тестирование программных систем, методы и инструментальные средства разработки программных систем.

Представлены результаты выполнения проектов, поддержанных грантами РФФИ (проекты № 18-01-00359, № 18-37-00070) и научного фонда НИУ ВШЭ (проект № 17-05-0020).

Материалы сборника могут представлять интерес для научных работников, специалистов в области программной инженерии, разработчиков информационных систем различного назначения, студентов и аспирантов, изучающих информационные технологии.

УДК 004.4+004.8+004.9

ББК 32.973

Печатается по решению учёного совета механико-математического факультета Пермского государственного национального исследовательского университета

Редакционная коллегия: А. И. Миков (КубГУ, Краснодар) – главный редактор, Л. Н. Лядова (НИУ ВШЭ – Пермь) – заместитель главного редактора, А. Ю. Гордилов (ПГНИУ, Пермь) – ответственный за выпуск, Ю. В. Кольцов (КубГУ, Краснодар), Е. Б. Замятина (НИУ ВШЭ – Пермь), Н. Н. Дацун (ПГНИУ, Пермь).

ISBN 978-5-7944-3237-4 (вып. 15)

ISBN 978-5-7944-1741-8

© ПГНИУ, 2018

© НИУ ВШЭ – Пермь, 2018

© КубГУ, 2018

А.Ю. Городилов, Р.А. Макурин

Пермский государственный национальный исследовательский
университет

aleksey.gorodilov@gmail.com, mak-jr@ya.ru

ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ В СИСТЕМЕ АВТОМАТИЧЕСКОГО СОСТАВЛЕНИЯ УЧЕБНЫХ РАСПИСАНИЙ

В статье описан подход к составлению учебного расписания в школе на основе генетических алгоритмов. Требования к расписанию разделены на две категории: критические, невыполнение которых влечёт появление некорректного, невыполнимого расписания, и пользовательские, позволяющие учесть индивидуальные предпочтения преподавателей и особенности конкретной школы. Критические требования к расписанию описываются на языке теории графов, что позволяет применять стандартные генетические операторы без порождения большого количества нежизнеспособных особей. Пользовательские требования описываются в виде гибко настраиваемой функции приспособленности, что делает возможной быструю адаптацию предложенного алгоритма к условиям конкретного учебного заведения. Алгоритм реализован в виде программы составления учебного расписания. В статье приведены результаты тестирования программы на реальных данных, показана корректность и преимущества предложенного подхода, отмечены пути развития.

Ключевые слова: генетические алгоритмы, составление расписаний, графы, функция приспособленности.

Введение

Проблема составления расписания учебных занятий является одной из наиболее актуальных задач в планировании и оптимизации процесса обучения, поскольку определяет эффективность работы преподавателей, усвоения учебного материала учениками, рациональность использования материальных ресурсов. Процесс составления расписа-

ния основан на анализе большого количества информации и требует значительных трудозатрат, поэтому в современных условиях возникает необходимость использования автоматизированных средств планирования и составления расписания в учебных заведениях.

Задача составления учебного расписания не может быть решена точными алгоритмами за приемлемое время, поскольку относится к числу NP-полных задач, а пространство поиска очень велико. Кроме того, четкие критерии оптимальности чаще всего отсутствуют, поэтому на практике для решения этой задачи применяют различные приближенные, эвристические алгоритмы. Некоторые из них показывают хорошие результаты, другие оказываются менее пригодны [1]. В данной статье для построения расписания предлагается использовать подход на основе генетических алгоритмов [2], успешно применяющийся для решения многокритериальных задач, а также задач, где критерии оптимальности заданы нечетко.

Из общения с лицами, ответственными за составление расписаний, можно сделать вывод, что имеющиеся на сегодняшний день программные средства для составления расписаний (такие, как «ХроноГраф» [3], «aSc Расписание», «1С:Автоматизированное составление расписания» [4], «АВТОРасписание» [5]) обладают рядом недостатков, основным из которых является невозможность учесть некоторые индивидуальные предпочтения или особенности конкретной школы. По этой причине диспетчеру по расписанию приходится в «ручном» режиме (в некоторых случаях даже на бумаге) корректировать составленное программным средством расписание, что снижает эффективность работы. Также диспетчеры отмечают неудобную структуру просмотра расписания и перегруженность интерфейса программ.

Таким образом, актуальность темы статьи обусловлена ростом требований к качеству обучения, планированию работы учеников, необходимостью рационального использования аудиторного фонда и учета дополнительных параметров оптимизации, а также недостатками существующих систем автоматического составления расписаний.

Формализация решаемой задачи

Введем некоторые ограничения в решаемую задачу. Будем считать, что расписание составляется на одну неделю («базовое расписание»), учебная неделя состоит из 5 дней, а каждый день у каждого класса должно быть не более 6 уроков. Таким образом, количество уроков в расписании одного класса не может быть больше 30.

Также введем некоторые основные определения, которые будем использовать в дальнейшем.

1. Класс – учебная группа, при составлении расписания будет являться минимальной неделимой единицей (деление класса на группы в рамках данной статьи рассматриваться не будет).

2. Позиция расписания – информация о времени проведения занятия, включает в себя день недели и порядковый номер урока. Как было указано выше, в данной статье будем считать, что всего возможно 30 различных позиций расписания.

3. Недельный план – документ, определяющий полный перечень предметов, изучаемых в учебном заведении, а также для каждого предмета количество часов в неделю и фамилию преподавателя в каждом классе; составляется на основе утвержденного учебного плана.

4. Урок – совокупность информации о времени (позиции расписания), классе, предмете и преподавателе. В понятие урока можно также включить информацию об аудитории (месте проведения занятия), однако в данной статье вопросы распределения аудиторий рассматриваться не будут, поэтому данная информация в понятие урока не включается.

5. Учебное расписание – последовательность уроков, множество которых соответствует недельному плану, при этом уроки не должны пересекаться по классам, преподавателям и аудиториям. Обычно расписание задается в табличном виде, где строки соответствуют времени, а столбцы – классу (расписание для учеников), преподавателю (расписание преподавателей) или аудитории (расписание занятости кабинетов).

Формализуем последние определения. Пусть C – множество классов, P – множество позиций расписания, T – множество преподавателей, S – множество предметов. Тогда каждый урок u – это кортеж $u = (p, c, t, s)$, где $p \in P, c \in C, t \in T, s \in S$, а учебное расписание – это отношение R на множестве $P \times C \times T \times S$. Недельный план, по сути, задает список кортежей (c, t, s) , где $c \in C, t \in T, s \in S$. Для построения расписания необходимо каждому такому кортежу поставить в отношение позицию расписания $p \in P$.

Пусть дано некоторое расписание R . Сформулируем два критических требования к расписанию.

$$1. \forall c_0 \in C, \forall p_1, p_2 \in P: (p_1, c_0, t, s) \in R, (p_2, c_0, t', s') \in R \Rightarrow p_1 \neq p_2. \quad (1)$$

$$2. \forall t_0 \in T, \forall p_1, p_2 \in P: (p_1, c, t_0, s) \in R, (p_2, c', t_0, s') \in R \Rightarrow p_1 \neq p_2. \quad (2)$$

По смыслу требование (1) соответствует отсутствию пересечений по классам, то есть требованию, что любые два урока в одном классе должны проходить в разное время, а требование (2) – отсутствию пересечений по преподавателям, то есть требованию, что любые два урока одного преподавателя должны проходить в разное время.

Остальные требования к расписанию будем называть пользовательскими. Примерами пользовательских требований являются: отсутствие «окон» у учеников и преподавателей, индивидуальные предпочтения преподавателей (отсутствие первых или последних уроков, наличие свободных дней и т.п.), чередование естественно-математических и гуманитарных предметов и т.д.

Будем говорить, что расписание R *корректно*, если оно удовлетворяет условиям (1) и (2). Будем говорить, что расписание *оптимально*, если оно корректно и удовлетворяет наибольшему числу пользовательских требований.

Таким образом, рассматриваемая задача заключается в нахождении оптимального расписания на основе заданного списка кортежей (c, t, s) , где $c \in C, t \in T, s \in S$, представляющего недельный план. Отметим, что при заданном недельном плане мощности всех корректных расписаний будут одинаковы и равны количеству заданных кортежей.

Задача составления учебного расписания относится к общей теории расписаний. Ее можно классифицировать по типу целевой функции как задачу многокритериальной оптимизации, критерии которой могут меняться с течением времени.

Кодирование решений в генетическом алгоритме

Будем использовать модель классического генетического алгоритма (ГА), поскольку именно для этой модели известны наиболее проработанные доказательства корректности работы алгоритма.

Наиболее очевидным является кодирование расписания в виде хромосомы, когда отдельными генами являются отдельные элементы отношения. Однако такой способ кодирования ведет к образованию большого числа возможных значений генов, что, в свою очередь, приводит к огромной мощности множества различных хромосом. При этом большинство из них будут соответствовать некорректным расписаниям в силу нарушения требований (1) и (2). Специально сгенерировать хромосому, одновременно удовлетворяющую указанным требованиям, достаточно сложно. Но даже если такие хромосомы будут сгенерированы, стандартные операторы скрещивания и мутации снова с высокой вероятностью приведут к нарушению этих условий. Учитывая указанные проблемы, будем использовать для кодирования промежуточное представление.

Представим недельный план в виде графа $G = (V, E)$, где V – множество кортежей (c, t, s) , $c \in C, t \in T, s \in S$. Пусть $v_i = (c_i, t_i, s_i)$, $v_j = (c_j, t_j, s_j)$. Ребро $(v_i, v_j) \in E \Leftrightarrow (c_i = c_j) \vee (t_i = t_j)$, где \vee – логическая функция дизъюнкции.

юнкции. То есть смежными вершинами будут являться уроки, в которых совпадает преподаватель или класс.

Правильно раскрасим граф G , при этом любые две смежные вершины окажутся окрашенными в разные цвета. Назначим каждому цвету некоторую позицию расписания $p \in P$ (например, цвет 1 – это первый урок в понедельник, цвет 2 – второй урок в понедельник, цвет 7 – первый урок во вторник и т.д.). В результате получим, что каждому кортежу $v_i = (c_i, t_i, s_i)$ поставлено в соответствие некоторое значение $p(i)$. Тогда множество кортежей $(p(i), c_i, t_i, s_i)$ соответствует расписанию R . Докажем, что такое расписание является корректным.

Действительно, если для $v_i = (c_i, t_i, s_i)$ и $v_j = (c_j, t_j, s_j)$ $c_i = c_j = c_0$, то по построению в графе G эти вершины соединены ребром. Следовательно, по определению правильной раскраски $p(i) \neq p(j)$, то есть требование (1) выполняется. Аналогично, если для $v_i = (c_i, t_i, s_i)$ и $v_j = (c_j, t_j, s_j)$ $t_i = t_j = t_0$, то по построению эти вершины тоже соединены ребром. Следовательно, $p(i) \neq p(j)$, то есть требование (2) выполняется. Таким образом, в результате правильной раскраски получено корректное расписание.

Для правильной раскраски графа применим «жадный» алгоритм, суть которого заключается в последовательном просмотре всех вершин в определенном порядке и раскраске каждой вершины в первый возможный цвет. Естественно, что получаемая раскраска существенно зависит от последовательности просмотра вершин. Хромосома в ГА как раз и будет кодировать такую последовательность. Таким образом, любая особь в ГА будет кодировать корректное расписание.

Функция приспособленности будет определять соответствие полученного расписания пользовательским требованиям. Чем больше и чем точнее выполнены пользовательские требования, тем больше значение функции приспособленности. Например, чтобы выполнить указанное выше требование, что количество позиций расписания не может быть больше 30, необходимо учитывать количество задействованных в раскраске цветов. Если их оказалось больше 30, то такая особь должна получить низкую приспособленность.

Поскольку ГА никак не зависит от внутреннего устройства функции приспособленности, становится возможным гибко настраивать критерии оптимальности, меняя фитнес-функцию, но абсолютно не изменяя весь остальной алгоритм.

Операторы и параметры генетического алгоритма

В реализованном генетическом алгоритме для отбора особей был выбран рулеточный отбор, при котором более приспособленная особь

имеет большую вероятность на скрещивание. Для скрещивания выбран односточный кроссовер, так как в решаемой задаче именно он позволяет потомку частично наследовать от родителей порядок раскраски. Для оператора мутации опытным путём установлено следующее правило: с вероятностью 5% происходит обмен четырех генов потомка между собой. Новое поколение формируется из всех полученных особей путём селекции.

Результаты

Для тестирования реализованного алгоритма при оценке оптимальности получаемого расписания использовалась функция приспособленности, построенная с учётом мнения экспертов – школьных диспетчеров по расписанию школ г. Добрянки и г. Перми. Для вычисления значения функции применяется следующий алгоритм.

1. $M \leftarrow 0$ (текущая оценка расписания)
2. Для каждого класса c_i , для каждой позиции расписания p_j :
 - 2.1. Если встретилось «окно» в середине дня, то $M \leftarrow M - 30$, иначе за каждый урок $M \leftarrow M + 1$.
 - 2.2. Если чередуются естественно-математические и гуманитарные предметы, то $M \leftarrow M + 1$.
3. Для каждого преподавателя t_i , для каждой позиции расписания p_j :
 - 3.1. Если встретилось «окно» в середине дня, то $M \leftarrow M - 30$, иначе за каждый урок $M \leftarrow M + 1$.

Недельный план для тестирования был взят из реальной школы. Он включал 15 классов, 29 учителей, 438 уроков. У каждого класса нагрузка составила по 28-30 уроков в неделю.

Размер популяции ГА составил 100 особей. Средняя оценка начальной популяции составляла около 300, при этом в среднем примерно 10% особей начальной популяции раскрашиваются успешно в 30 цветов. При выполнении генетического алгоритма средняя оценка популяции растёт, а разнообразие падает, и мы достигаем локального максимума функции. Получаемое в результате работы ГА расписание не содержит «окон» (рис. 1), что как раз и отражено экспертами в фитнес-функции как главный критерий. Стоит отметить, что даже если в начальной популяции не было особи, которая раскрашивается в 30 цветов, в процессе генетического алгоритма с высокой вероятностью такая особь появлялась.

		5А	5Б	5В
понедельник	1	География Тунчак С.К.	Технология Тоттава Н.И.	Биология Логун Е.М.
	2	Математика Климов М.Т.	Физкультура Мордлина Г.А.	География Тунчак С.К.
	3	Русс/литер Иванов В.В.	Технология Тоттава Н.И.	Музыка Кринчук М.В.
	4	Иностранный яз. Петров К.С.	Математика Климов М.Т.	Русс/литер Иванов В.В.
	5	Математика Климов М.Т.	Иностранный яз. Петров К.С.	История/общество. Гаврилова О.И.
	6	ИЗО Окулова С.С.	Русс/литер Иванов В.В.	Математика Климов М.Т.

Рис.1. Фрагмент сгенерированного расписания

Динамика работы ГА отражена на рис. 2, который демонстрирует высокую скорость и монотонный характер роста функции приспособленности, что подтверждает корректность реализованного алгоритма.

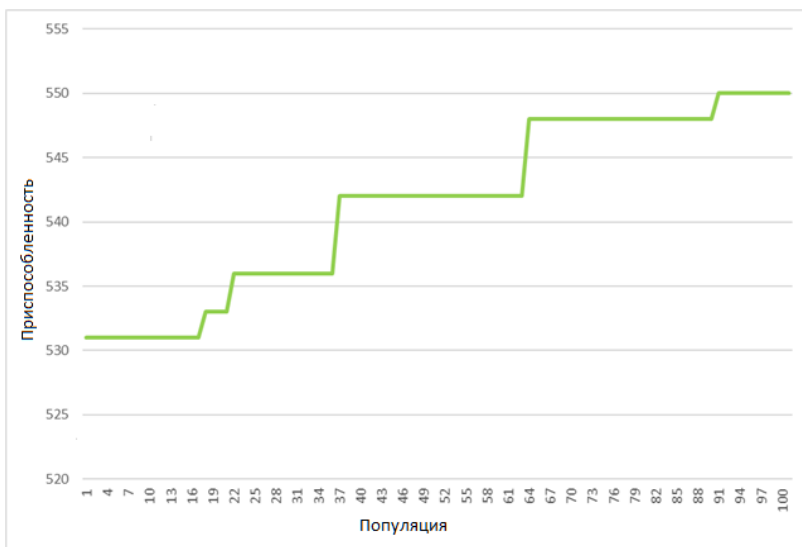


Рис. 2. График изменения значения функции приспособленности лучшей особи в популяции

При использовании тривиального способа генерации расписания для четырех классов одной параллели лишь 1% из 1000 сгенерированных расписаний получились корректными.

Заключение

Разработанная программа позволяет автоматически составлять учебное расписание в соответствии с введенным недельным планом, установленными ограничениями и предпочтениями со стороны преподавателей. Тестирование проведено на реальных данных с учетом мнения экспертов. Анализ результатов показал высокую скорость и монотонный характер роста функции приспособленности, что подтверждает корректность реализованного алгоритма. Новизна предложенного генетического алгоритма заключается в использовании нестандартного кодирования, которое позволяет избежать образования большого количества нежизнеспособных особей, соответствующих некорректным расписаниям. Сравнение с тривиальным способом представления расписаний продемонстрировало высокую эффективность выбранного способа.

В настоящее время на уровне интерфейса в системе реализована возможность задания лишь небольшого числа пользовательских требований (рис. 3).

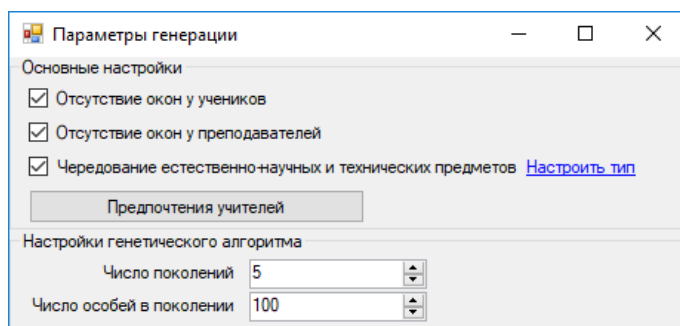


Рис. 3. Форма настроек пользовательских требований и параметров ГА

Однако, как отмечалось выше, архитектура приложения даёт возможность гибкой настройки критериев оптимальности. Хоть такая настройка и не требует внесения изменений в основной программный код приложения, все же требуется изменение исходного кода, отвечающего за вычисление фитнес-функции. Очевидно, что рядовые пользователи (диспетчеры, составляющие расписание), не обладают необходимой для этого компетенцией. Таким образом, основным направлением дальнейшего развития является доработка пользовательского интерфейса для изменения критериев оптимальности. Кроме того, планируется предоставление пользователю возможности изменения

расписания вручную с помощью удобных визуальных инструментов. Также интерес представляет разработка DSL для максимально гибкого задания пользовательских требований.

Библиографический список

1. The Second International Timetabling Competition (ITC-2007): Curriculum-based Course Timetabling [Электронный ресурс] URL:<http://www.cs.qub.ac.uk/itc2007/curriculumcourse/report/curriculumtechreport.pdf>
2. Mitchell M. An Introduction to Genetic Algorithms / M. Mitchell. – Cambridge: The MIT Press, 1996.
3. ХроноГраф 3.0 Мастер [Электронный ресурс] URL: http://www.chronobus.ru/normbase/detail.php?ELEMENT_ID=1772564
4. 1С:Автоматизированное составление расписания. Школа [Электронный ресурс] URL: <https://solutions.1c.ru/catalog/timetable/features>
5. АВТОРасписание [Электронный ресурс] URL: <https://www.mmis.ru/programs/avtor>

А.П. Даденкова

Пермский национальный исследовательский
политехнический университет

dadenkovaa@yandex.ru

ИССЛЕДОВАНИЕ ХАРАКТЕРИСТИК МОДЕЛИ ПРОТОКОЛА ПОИСКОВОГО СЕРВИСА DHT

В данной статье выполняется исследование модели децентрализованной файлообменной сети с протоколом DHT и предлагаются рекомендации по выбору значений параметров. В работе кратко описаны принципы функционирования поискового протокола, предложены фрагменты реализации его модели в среде имитационного моделирования AnyLogic. Проведены и представлены результаты экспериментов по исследованию протокола и выполнен анализ результативности поиска данных файлообменной сети.

Ключевые слова: имитационное моделирование, протоколы поиска данных, децентрализованные файлообменные сети, AnyLogic, DHT, P2P.

Современный этап построения файлообменных сетей характеризуется широким применением технологии peer-to-peer (P2P). Это связано с постоянным увеличением масштабов сетей, что в клиент-серверной архитектуре приводит к снижению отказоустойчивости и пропускной способности сети. Распространённость получают децентрализованные системы хранения и протоколы поиска данных, и в частности, протокол поискового сервиса DHT (Distributed Hash Table) [1, 2], получивший применение в таких сетях как BitTorrent, I2P, ONEnet и других.

Актуальность проблемы использования протокола DHT в файлообменных сетях обоснована недостаточным количеством исследований влияния его параметров на результативность поиска данных в сети, что приводит к нерациональному использованию пропускной способности каналов передачи и снижению эффективности поиска данных [3]. Таким образом подтверждается важность исследования вопросов постро-

ения имитационной модели файлообменной сети с протоколом DHT и анализа влияния его параметров на результативность поиска данных в сети.

Протокол DHT предназначен для организации поиска данных путём создания уникальных индексов всех имеющихся файлов в распределённых P2P-сетях. Особенность протокола заключается в том, что каждый участвующий узел имеет собственную хеш-таблицу ассоциативных записей идентификаторов и адресов других узлов сети и с помощью неё может найти искомые данные, связанные с идентификаторами других узлов. Информация распределяется среди некоторого набора узлов таким образом, чтобы за несколько итераций каждый узел смог найти необходимые данные. При этом узлы самостоятельно поддерживают связи между собой, что позволяет легко масштабировать сеть.

В основе протокола находятся процедуры подключения пользователей, поиска, размещения и распространения информации в сети.

Подключение нового пользователя к сети включает [4]:

1) выбор уникального 160-битного идентификатора hashID (используется хеш-функция SHA-1);

2) заполнение и сортировка (по возрастанию) DHT-таблицы случайными существующими в сети идентификаторами узлов и соответствующими адресами (IP:port).

Записи в DHT-таблицу добавляются в несколько итераций до тех пор, пока не будет достигнут минимальный размер таблицы v . Каждая итерация представляет собой запрос к DHT-таблице случайно выбранного узла сети для получения записей из корзины таблицы (k случайных записей, первая из которых является ключом корзины) и добавление полученных идентификаторов в заполняемую таблицу. При этом только первая итерация является обращением к серверу-координатору, а все последующие – к добавленным в таблицу узлам.

Далее, на основе уникального идентификатора раздачи данных infoHash, происходит добавление и поиск данных в сети. Размещение данных состоит в осуществлении поиска m представителей данных, которые будут знать о наличии у него раздаваемой информации. Процедура поиска состоит в нахождении в сети узлов с идентификаторами hashID наиболее близкими к infoHash искомым данным.

Поиск представителей производится за q итераций, называемых глубиной поиска. Итерации аналогичны тем, что производятся при заполнении DHT-таблицы, кроме адресата запроса и искомой информации. Каждая итерация включает:

1) поиск в DHT-таблице узла корзины идентификаторов наиболее

- близкой (наименьшее расстояние) к искомому infoHash;
- 2) формирование запроса к узлам корзины на получение от каждого из них k наиболее близких DHT-записей к искомому infoHash;
 - 3) добавление новых записей в таблицу и сортировка.

Процедура завершается выбором из DHT-таблицы узлов представителей и передачи им идентификатора узла раздающего данные и infoHash его данных.

Процедура поиска данных состоит в итеративном обращении к узлам и заполнению DHT-таблицы потенциально новыми идентификаторами узлов, чьи hashID ближе всего к искомому infoHash. Во время поиска опрашиваемые узлы сообщают пользователю о наличии у них искомой информации.

Построение имитационной модели файлообменной сети выполняется в системе объектного дискретно-событийного моделирования AnyLogic. Система моделирования позволяет учесть значимые параметры узлов сети, что обеспечивает корректность результатов моделирования.

Архитектура разработанной модели сети [4] имеет иерархическую структуру. Объектом верхнего уровня является сеть «DHT_Network», а вложенными: узлы класса «Node», сервер «Server». Элементами объекта «Сеть» являются: arrayNodes – популяция узлов сети; server – сервер файлообменной сети; numberNodes – количество узлов в сети; nodeInfo – узел сети, раздающий данные; infoHash – хэш искомых в сети данных; depthSearch – глубина поиска infoHash (число итераций); sizeBucket – размер корзины DHT-таблицы; nodeSearch – узел сети, ищущий данные; numberCycles – количество прогонов модели; creatingNodesAndHashID – функция создания узлов сети и назначения им уникальных hashID; fillingTableDHT – функция заполнения и сортировки DHT-таблиц; createInfoHash – функция выбора раздающего узла и создания infoHash искомых данных; searchIndexBucket – функция поиска индекса корзины DHT-таблицы узла node, ближайшей к infoHash; searchDelegates – функция назначения узлов-представителей infoHash в сети; nodeSelection – функция выбора узла, выполняющего поиск данных; searchData – функция поиска данных. Элементами объекта «Узел» сети являются: hashID – хэш узла; DHT_Table – таблица хэш'эй узлов сети; minNumberEntries – минимальное число записей в таблице DHT; infoHash – хэш данных узла. Сервер содержит только коллекцию arrayIDs ассоциативных записей hashID и индексов соответствующих узлов сети.

Реализация алгоритма хеширования SHA-1 в модели осуществляется методами библиотеки org.apache.commons.codec.digest.DigestUtils

подключённого распространённого пакета commons-codec-1.10.jar.

Для исследования модели протокола DHT и анализа результативности поиска принимаются следующие исходные данные: количество узлов в сети фиксировано и равно 50000, количество записей в таблице фиксировано - 1000, значение глубины поиска варьируется в диапазоне [1..8] (эксперимент №1), значение размера корзины изменяется от 1 до 8 (эксперимент №2). Оценка эффективности работы протокола выполняется по характеристикам результативности нахождения данных и итоговой общесетевой нагрузке. Нагрузка на сеть определяется количеством поисковых запросов и ответов:

$$U = q \times (k + k^2). \quad (1)$$

Первый эксперимент состоит в изучении влияния глубины поиска на вероятность нахождения данных в сети (см. рис. №1). Моделирование проводится для значений размера корзины 4 и 8. Результаты оценки характеристик модели продемонстрированы на рис. 1.

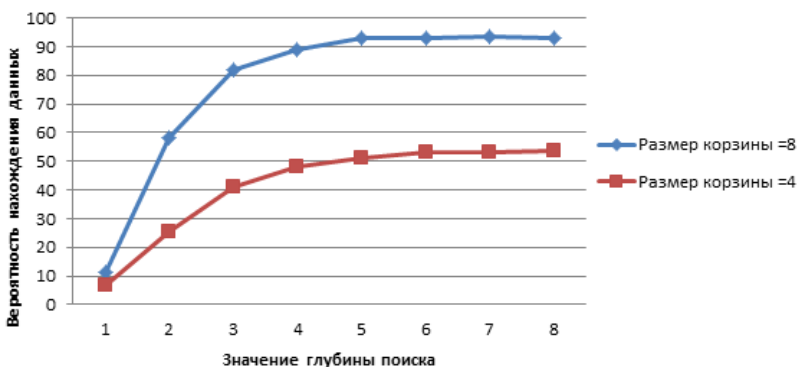


Рис. 1. График зависимости результативности поиска от глубины

График содержит два различных участка изменения вероятности нахождения данных в сети: первый участок (при значении глубины поиска от 1 до 5) характеризуется экспоненциальным ростом результативности поиска, а второй (глубина поиска >5) – отсутствием изменений. Результаты характерны для произвольного размера корзины, что подтверждается моделированием. Это позволяет отметить рациональное значение глубины поиска, увеличение которого приводит к значительному росту нагрузки и малому влиянию на результативность поиска. Можно предположить, что рациональная глубина поиска определяется соотношением числа записей в DHT-таблице и числа уз-

лов в сети (в рассматриваемом примере 1000/50000).

Второй эксперимент состоит в анализе влияния размера корзины на сетевую нагрузку и результативность поиска (см. рис. 2).

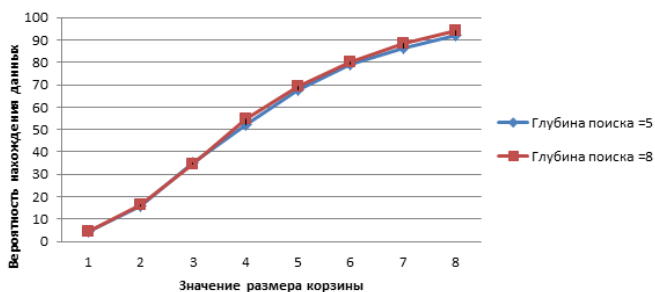


Рис. 2. График зависимости результативности поиска от размера корзины

Моделирование проводится при значениях глубины поиска 5 и 8 для подтверждения ранее сформулированной гипотезы рационального выбора глубины поиска. Результаты моделирования представлены в табл. 1 и на рисунке 2.

Таблица 1. Зависимость нагрузки на сеть от размера корзины и глубины поиска

Размер корзины	1	2	3	4	5	6	7	8
Глубина поиска =5	10	30	60	100	150	210	280	360
Глубина поиска =8	16	48	96	160	240	336	448	576

Сравнение результативности поиска в сети при различной глубине поиска показывает следующие результаты:

- 1) значение глубины поиска выше рационального практически не влияет на вероятность нахождения данных;
- 2) значение глубины поиска оказывает сильное влияние на сетевую нагрузку;
- 3) увеличение размера корзины сопровождается линейным изменением результативности поиска.

Сформулированная гипотеза рационального выбора глубины поиска подтверждается, при этом данные эксперимента позволяют говорить о рациональном значении размера корзины. Учитывая возрастающую нагрузку на сеть, более эффективным является выбор меньшей глубины поиска. Таким образом, результаты эксперимента позволяют управлять сетевой нагрузкой изменением значения глубины поиска.

Дальнейшее исследование поискового протокола DHT предполагает выполнение количественной оценки и анализа результативности поиска информации в сети с различным числом узлов, записей DHT-таблицы, глубиной поиска и размером корзины. Конечной целью анализа является оптимизация характеристик поискового протокола для заданных параметров сети.

Библиографический список

1. Loewenstern A. DHT Protocol [URL: http://www.bittorrent.org/beps/bep_0005.html].
2. Рабинович В.Е. Способ управления трафиком в BitTorrent-сетях с помощью протокола DHT / В.Е. Рабинович, А.А. Шестаков // Вестник СибГУТИ. – 2012. – №3. – С. 3-10
3. Сиващенко Д. Распределенные хэш-таблицы на примере NoSQL / Д. Сиващенко // [Режим доступа: <http://seminar.at.ispras.ru/wp-content/uploads/2010/11/Cassandra-and-DHT.pdf>].
4. Даденкова А.П. Построение модели файлообменной сети с протоколом DHT / А.П. Даденкова // Научно-технический вестник Поволжья. – 2017. - № 6. – С. 183-186.

Е.Б. Замятина, А.В. Кудрявцев ¹

Национальный исследовательский университет
«Высшая школа экономики»

Пермский государственный национальный
исследовательский университет

ezamyatina@hse.ru

МОДЕЛИРОВАНИЕ АЛГОРИТМОВ МАРШРУТИЗАЦИИ AD-HOC-СЕТЕЙ С ИСПОЛЬЗОВАНИЕМ АГЕНТНЫХ ТЕХНОЛОГИЙ

В работе рассматриваются различные классы ad-hoc-сетей, алгоритмы маршрутизации, более подробно исследуются алгоритмы маршрутизации сетей FANET. Приведено краткое описание системы имитационного моделирования TriadNS, с помощью которой построена имитационная модель ad-hoc-сети. Обосновывается целесообразность применения агентной парадигмы имитационного моделирования для исследования сетей.

Ключевые слова: имитационное моделирование, агентная технология, протоколы маршрутизации, ad-hoc-сети.

Введение

Благодаря прогрессу в беспроводных технологиях появилось большое количество устройств (GPS, Wi-Fi интерфейсов, компьютеров со встроенными микропроцессорами и т.д.), используемых в повседневной жизни.

Вслед за этими приспособлениями появились достаточно миниатюрные «умные» беспилотные летающие аппараты, что привело к появлению сетей нового вида, несущих название FANET (fly ad hoc network)[1]. С внедрением FANET, возникли различные виды гражданских и военных приложений, предназначенных для координации спасательных команд на земле, пограничного контроля, для кон-

¹ Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 18-01-00359

троля сельскохозяйственных угодий, обнаружение нефтяных месторождений и т.д.

Наряду с сетями FANET, развиваются сети VANET (vehicular ad hoc network)[2]. Эти сети предполагают осуществить взаимодействие транспортных средств между собой и с сетью связи общего пользования. Необходимо в автомобиле создавать сетевой интерфейс, который позволил бы поддерживать такие группы соединений, как автомобиль-автомобиль, автомобиль-инфраструктурная сеть, автомобиль-жилье, маршрутизация (архитектура сети VANET предполагает взаимодействие автомобиля, как с другими автомобилями, так и с придорожной сетью). VANET превращает автомобиль в беспроводной узел сети или маршрутизатор.

И для той и для другой сети характерно то, что они могут функционировать без телекоммуникационных операторов (без их инфраструктуры). Все узлы сети взаимодействуют в процессе передачи сообщения. Следует отметить подвижность узлов сети, а эти сети называют “мобильными”[3] (MANET-Mobile Ad hoc Network), сети FANET и VANET являются подклассами мобильных сетей.

Итак, задача мобильных сетей - осуществить передачу данных в сети по определённом алгоритму маршрутизации. Существует большое количество алгоритмов маршрутизации для мобильных сетей, которые мы рассмотрим ниже. Алгоритмы учитывают особенности мобильных сетей различных классов. В этой статье мы более подробно рассмотрим класс мобильных сетей FANET.

При разработке алгоритмов маршрутизации зачастую используют метод имитационного моделирования. Необходимость в моделировании алгоритмов объясняется тем, что при их разработке необходимо добиться достаточно высокой производительности надёжности этих алгоритмов.

Статья включает следующие разделы: первый раздел посвящён различным алгоритмам маршрутизации FANET, во втором разделе рассматриваются система имитационного моделирования TriadNS, далее – лингвистические конструкции языка Triad, которые используют для описания программного агента и результаты тестирования.

Алгоритмы маршрутизации в сети FANET

Существует два типа приложений, работающих с воздушными узлами: (а) одноузловое приложение, в котором воздушный узел находится в центре множества базовых станций, находящихся на земле (узел может быть использован станциями в качестве роутера для коммуникации с другими станциями, находящимися вне их поля диапазона

связи); (б) многоузловое приложение [1].

Если мы используем один узел, то может возникнуть проблемы, а именно: (а) узел может обладать крайне небольшим диапазоном связи, (б) могут возникать помехи в процессе передачи сигнала. По этой причине для передачи данных используют группу воздушных узлов. Преимущества многоузловых приложений заключаются в следующем: (а) многоузловое приложение способствует повышению отказоустойчивости (при выходе из строя одного узла другие узлы берут на себя его работу); (б) отдельные задачи могут выполняться параллельно различными узлами, что в значительной мере уменьшает продолжительность выполнения всего задания в целом. Необходимо учитывать, что для надежной работы алгоритмов маршрутизации необходимо достаточное количество узлов.

Надёжная маршрутизация является важным фактором стабильности приложений и услуг, которые они предоставляют. Существует широкий спектр протоколов маршрутизации, предложенных для FANET. Все эти протоколы призваны улучшить коэффициент доставки пакетов и минимизировать задержки передачи данных, а также, вероятность потери пакетов.

Протоколы маршрутизации FANET можно разделить на три основные категории: (а) протоколы маршрутизации на основе топологии, (б) роевые (Swarm) протоколы маршрутизации, (в) протоколы маршрутизации на основе географического местоположения.

Протоколы маршрутизации, основанные на топологии (скорость перемещения узлов в воздухе невысока, число узлов ограничено) недостаточно изучены для FANET. Дело в том, что, если скорость перемещения узлов высока, то алгоритмы, основанные на топологии, не могут быть применены достаточно успешно. Ещё одним недостатком этих алгоритмов является потребление большего количества ресурсов и энергии (относительно большие затраты памяти, например, для хранения таблиц маршрутизации). Итак, категория протоколов маршрутизации, основанная на топологии, использует IP-адреса для определения узлов и использует существующую информацию о ссылках в сети для пересылки пакетов по соответствующему пути. Протоколы классифицируются следующим образом: проактивная маршрутизация (протоколы OLSR, D-OLSR, M-OLSR, CE-OLSR и DSDV), реактивная (AODV, AODV-SEC, AODV с временным интервалом, M-AODV, APAR и DSR) и гибридная (HWMP, ZRP, SHARP, HRPO и TORA). Гибридная категория протоколов предполагает, что сеть разделяется на зоны, в отдельной зоне принимается проактивная маршрутизация, а связь между зонами основана на реактивной маршрутизации.

Роевые протоколы маршрутизации основаны на роевом интеллекте (SI), который впервые использовался для клеточной роботизированной системы. Роевые алгоритмы моделируют социальное поведение птиц или рыб у стада или насекомых на рое (выше перечислены примеры SAC – самоорганизующихся адаптивных систем). Алгоритмы SI направлены на поиск оптимального решения. В качестве протоколов маршрутизации на основе роя, которые предназначены для FANET, можно выделить BeeAdhoc и APAR.

Протоколы, основанные на определении географического местоположения: этот класс протоколов маршрутизации основан на знании географического местоположения узлов сети, которое каждый узел может определить с помощью GPS.

Для оценки производительности протоколов маршрутизации используют, как правило, шесть основных показателей (для FANET):

1. Коэффициент доставки пакетов (PDR): определяется как отношение успешно доставленных пакетов в пункт назначения ко всем пакетам данных, сгенерированным отправителями. Чем больше PDR, тем лучше производительность протокола.

2. Средняя конечная задержка (EED): среднее время прохождения пакетов данных до целевых пунктов назначения. Чем меньше EED, тем лучше производительность протокола.

3. Среднее количество переходов (H): определяется как количество доставленных пакетов данных, делённое на количество переходов, выполняемых всеми пакетами

4. Накладные расходы (O): отношение пакетов маршрутизации к успешно доставленным пакетам в приёмниках. Этот показатель показывает степень насыщения сети. Чем ниже, тем лучше.

5. Пропускная способность (T): количество пакетов данных, успешно доставляемых в целевой пункт назначения в течение заданного промежутка времени (обычно 1). Чем больше значение T, тем лучше производительность протокола маршрутизации.

6. Задержка (L): мера времени, затрачиваемого пакетом данных для транзита между двумя узлами в данной сети. Чем ниже значение L, тем лучше алгоритм маршрутизации.

Одним из основных методов анализа алгоритмов маршрутизации является имитационное моделирование. Существуют специальные системы имитации, предназначенные для моделирования компьютерных сетей. Дадим краткий обзор симуляторов компьютерных сетей.

Обзор симуляторов компьютерных сетей

В настоящее время существует достаточное большое количество симуляторов компьютерных сетей, в том числе: COMNET [4], OPNET [5], OMNeT++ [6]. Все они обладают привлекательными чертами, которые позволяют выполнять автоматизированное проектирование компьютерных сетей, получать временные оценки алгоритмов.

Так COMNET III – объектно-ориентированная система моделирования локальных и глобальных сетей. Позволяет моделировать уровни: приложений, транспортный, сетевой, канальный. Использует все известные на сегодня технологии и протоколы, а также системы клиент-сервер. Модель создаётся из объектов, “строительных блоков”, с COMNET поставляется большая библиотека объектов-моделей реального сетевого оборудования и методов доступа к среде.

OPNET – также представляет собой программное средство для проектирования и моделирования локальных и глобальных сетей, компьютерных систем, приложений и распределенных систем. Включает следующие программные продукты: Netbiz (проектирование и оптимизация вычислительной системы), Modeler (моделирование и анализ производительности сетей, компьютерных систем, приложений и распределенных систем), ITGuru (оценка производительности коммуникационных сетей и распределенных систем).

OMNeT++ представляет собой симулятор дискретных событий, которые происходят внутри простых модулей (simple modules). В системе OMNeT++ заложена детальная реализация протоколов, начиная с сетевого уровня, возможность написания и подключения собственных модулей, развитый графический интерфейс.

Можно сделать заключение, что описанные выше программные средства для проектирования, анализа и исследования компьютерных сетей являются низкоуровневыми. Ещё один недостаток: симуляторы являются коммерческими.

Наряду с использованием специальных средств имитационного моделирования используют системы моделирования общего назначения. Исследование компьютерных сетей можно провести с помощью развитых систем имитационного моделирования общего назначения, например, GPSS [7] или AnyLogic [8].

Однако эти системы моделирования лишены некоторых специализированных лингвистических и программных средств для описания структурных особенностей компьютерных сетей и функционирования узлов, а также для сбора статистических данных, которые могли бы определить производительность.

Система моделирования TriadNS и агентное моделирование

В TriadNS [9,10] с помощью специально разработанного высокоуровневого языка моделирования Triad можно описать новое устройство или, *структуру* компьютерной сети (математической моделью локальной компьютерной сети является граф, математической моделью ad hoc сетей, к которым относятся и сети FANET – динамические случайные графы).

Используя лингвистические [11,12] и программные средства *слоя рутин* СИМ TriadNS, есть возможность описать новый алгоритм. Нашей задачей является построение алгоритмов маршрутизации в ad hoc сетях.

Средства сбора в ходе имитационного эксперимента статистической информации о построенной имитационной модели позволяют проанализировать ее, предоставив в распоряжение исследователя результаты работы стандартных и пользовательских *информационных процедур*.

Условия моделирования языка Triad позволяют задать алгоритм исследования и, в результате, получить модель, соответствующую заданным критериям.

Известно, что существуют различные парадигмы имитационного моделирования: ориентированная на события, процессо-ориентированная, объектно-ориентированная, агентная.

Так авторы СИМ ANYLOGIC позиционируют ее как многоподходную, т.е. исследователь может применить как средства системной динамики, так и средства, реализующие процессо-ориентированное и агентное имитационное моделирование. Известно, что агентное имитационное моделирование предполагает, что имитационная модель представляет собой взаимодействующих программных агентов. Агенты могут быть реактивными (реагировать на внешние воздействия окружающей среды) или интеллектуальными, т.е. могут принимать решения.

Поскольку зачастую в алгоритмах маршрутизации ad hoc сетей возникают сложные ситуации и вычислительный узел должен принимать решения, менять своё поведение в зависимости от изменений во внешней среде, целесообразно использовать агентную парадигму имитационного моделирования.

В качестве описания действий агентов можно использовать рутинны языка TriadNS. Рассмотрим синтаксис рутинны. Каждую рутину задают множеством событий E , линейно упорядоченным множеством временных моментов (T) и набором состояний $\{Q\}$. Состояние задаётся

значениями локальных переменных. Локальные переменные определяются в рутине. Состояние изменяется только в случае возникновения события. События планируют друг друга (*schedule* (*event1*, *t1*) – процедура планирования события *event1* на момент времени *t1*). В рутине могут быть определены входные и выходные полюсы (Pr_{in} и Pr_{out}). Входной полюс служит для приёма сообщений, выходные полюсы – для их отправки. Среди множества событий выделяют входные сообщения. Все входные полюсы обрабатываются входным событием, выходные полюсы – обычным событием. Специальный оператор (*out* <сообщение> *through* <имя полюса>) служит для отправки сообщения.

Синтаксис рутины на языке Triad имеет следующий вид:

```
Routine <Имя> { <Секция параметров> |
                  <Объявление полюсов> }
          [ <Секция инициализации рутины> ]
          { <Описание события рутины> }
EndRout
```

Рассмотрим тестовую задачу поиска нужного вычислительного узла в сети.

В начальный момент времени агент находится в некоторой точке *A1*. Его цель – добраться до объекта *G*. Между этими объектами находятся другие вычислительные узлы, через которые пролегает путь.

Также изначально в задаче известны расстояния между связанными объектами. При этом агент, осуществляющий поиск, из альтернатив при выборе направления предпочтёт тот вычислительный узел, расстояние до которого меньше.

Граф, представляющий вычислительные узлы и связи между ними, представлен на рис. 1.

Ниже приведена рутина и агент, который осуществляет поиск нужного узла в компьютерной сети.

Рутина *G* – отдельная рутина, выделенная для поиска объекта:

```
routine G(inout Con[10])
  initial
    integer place;
    place:=8;
    schedule CheckSearching in 0;
  endi
  event CheckSearching
    if (mIndex = 8) then Print "Found!";
```

```

eor;
endf;
schedule CheckSearching in 1;
ende
endrout

```

В секции инициализации рутина содержит переменную *place*, являющуюся позицией (индексом среди вершин). Событие *CheckSearching*, следит за тем, найдена ли конечная вершина.

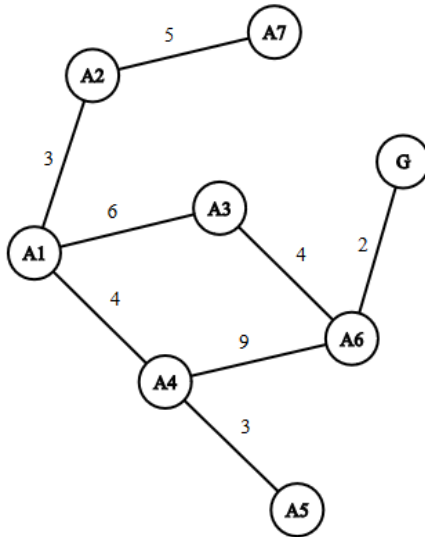


Рис. 1. Граф перемещений

Рутинa *G* реализует алгоритм функционирования конечного узла. Алгоритм, по которому функционируют остальные узлы, реализован с помощью другой рутинy. Рутинa содержит вспомогательные переменные (предварительно описанные в слое сообщений), которые позволяют хранить информацию, полученную в результате опроса ближайших узлов, определять расстояние до этих узлов, индексы узлов. Механизм вывода позволяет определить вычислительный узел с нужным индексом.

Были проведены соответствующие эксперименты, которые показали жизнеспособность разработанных программных средств.

Заключение

В работе рассмотрены ad hoc сети, рассмотрены особенности сетей, различные классы. Более подробно рассмотрен класс беспроводных сетей для беспилотных летательных аппаратов (FANET) и алгоритмы маршрутизации в этих сетях. Алгоритмы маршрутизации сетей FANET интересны тем, что узлы сети являются мобильными, а сети – многоузловыми, следовательно, являются достаточно сложными.

Для изучения и исследования алгоритмов применяют методы имитационного моделирования, наиболее адекватными в этом случае являются агентные технологии имитационного моделирования. В работе обоснована возможность применения СИМ TriadNS для исследования ad-hoc-сетей и приводится тестовый пример, в котором используют разработанные авторами программные средства, реализующие агентов и выполняющих механизм логического вывода, показывающие возможности подхода и разработанной системы.

Библиографический список

1. *Oubbatia O.S., Lakas A., Zhouc M., Güne M.B., Yagoubia F.* A survey on position-based routing protocols for Flying Adhoc Networks (FANETs)// *Vehicular Communications*, 10, Elsevier, 2017. P. 29–56.
2. *Boussoufa-Lahlaha S., Semchedinea F., Bouallouche-Medjkounea L.* Geographic routing protocols for Vehicular Ad hoc NETworks (VANETs): A survey// *Vehicular Communications*, 11, Elsevier, 2018. P. 20–31.
3. *Li M., Salmanian M., Willink T.J.* Network layer connectivity awareness with application to investigate the OLSR protocol in tactical MANETs // In: *Proceedings of the 2017 Winter Simulation Conference*. P. 4093–4105.
4. COMNET III. Planning for Network Managers. Release 1.3. [Электронный ресурс] [Режим доступа: <http://eent3.lsbu.ac.uk/staff/baoyb/acs/Comnet/comnet%20III.pdf>] [Проверено: 30.10.2016].
5. OMNeT++ Community Site. [Электронный ресурс] [Режим доступа: <http://www.omnetpp.org>] [Проверено: 30.11.2018].
6. OPNET. [Электронный ресурс] [Режим доступа: <http://www.riverbed.com/products/performance-management-control/opnet.html?redirect=opnet>] [Проверено: 30.11.2018].
7. GPSS. [Электронный ресурс] [Режим доступа: <http://www.gpss.ru>] [Проверено: 30.11.2018].

8. Anylogic [Электронный ресурс] [Режим доступа: <http://www.anylogic.ru>] [Проверено: 30.11.2018].
9. *Mikov A., Zamiatina E.* Program Tools and Language for Networks Simulation and Analysis // In: Proceedings SDN & NFV – The Next Generation of Computational Infrastructure: 2014 International Science and Technology Conference «Modern Networking Technologies (MoNeTec)», October 27-29, 2014, Lomonosov Moscow State University. P. 94-102.
10. *Zamyatina E., Mikov A, Mikheev R.* Linguistic and Program Tools For Debugging and Testing Of Simulation Models Of Computer Networks // International Journal "Information Models and Analyses". 2013. Vol. 2. No. 1. P. 70-80.
11. *Mikov A.I.* Simulation and Design of Hardware and Software with Triad // In: Proc. 2nd Intl. Conf. on Electronic Hardware Description Languages. Las Vegas, USA. 1995. P. 15-20.
12. *Миков А.И.* Автоматизация синтеза микропроцессорных управляющих систем – Иркутск: издательство Иркут. ун-та, 1987.

Е. Л. Кон, С. А. Даденков²

Пермский национальный исследовательский политехнический университет (ПНИПУ)

kel-40@yandex.ru, dadenkov@rambler.ru

АНАЛИЗ СТЕКОВ ПРОТОКОЛОВ FIELDBUS-СЕТЕЙ СО СЛУЧАЙНЫМ МНОЖЕСТВЕННЫМ ДОСТУПОМ CSMA

Выполняется анализ общности распространённых коммуникационных стеков протоколов сенсорных fieldbus-сетей с алгоритмами случайного множественного доступа CSMA узлов к разделяемой среде передачи. Проводится сопоставление стеков протоколов сетей с семиуровневой моделью взаимодействия открытых систем. Представляется краткая характеристика используемых в анализируемых сетях алгоритмов доступа и их различия. В результате анализа выделен наиболее сложный для исследования семиуровневый коммуникационный стек протоколов сети LonWorks, характеризующий параметрической и функциональной общностью со всеми анализируемыми стеками. Полученный результат позволяет выполнять дальнейшее обобщённое исследование сетей, результаты которого могут быть использованы для широкого круга сенсорных fieldbus-сетей со случайным доступом.

Ключевые слова: коммуникационный стек протоколов, промышленная шина, сенсорная сеть, CSMA, fieldbus, network, LonWorks.

Введение

Протоколы случайного множественного доступа с контролем несущей широко распространены и применяются в сенсорных fieldbus-сетях с разделяемой (общей) средой передачи данных. В настоящее время начитывается большое множество сетей, использующих различные протоколы случайного множественного доступа с контролем не-

² Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта №18-37-00070 «Алгоритмы проектирования информационно-управляющих сетей LonWorks с требуемыми вероятностными и временными характеристиками».

сушей, предотвращением и избеганием коллизий: CSMA/NBA (шины CAN, KNX/EIB, DeviceNet), 1-CSMA/CD (EtherNet/IP, ModBus/TCP), predictive *p*-Persistent CSMA (BacNet, LonWorks). Отличительными особенностями протоколов является обеспечение эффективности функционирования сети (времени и надёжности передачи) при различной загруженности каналов сети, количестве узлов, периодичности/спорадичности сетевой нагрузки и других параметрах.

Проектирование fieldbus-сетей требует обеспечения высоких показателей надёжности и низких временных затрат информационного обмена узлов (сенсоров) в сети. Проектирование осложнено высокими темпами появления и развития сетей, опережающими разработку новых инструментов для оценки характеристик сетей и построения их архитектуры. Отсутствие анализа схожести функционирования протоколов не позволяет модифицировать и использовать уже известный инструментарий для проектирования протокольно схожих сенсорных сетей. Это актуализирует проработку вопросов анализа общности распространённых коммуникационных стеков протоколов сенсорных fieldbus-сетей с алгоритмами случайного множественного доступа CSMA и выделения распространённого и наиболее сложного для анализа стека, перспективная проработка инструментария проектирования которого позволит использовать его для схожих сетей.

Для достижения поставленной цели в работе выполняется анализ распространённых коммуникационных стеков протоколов и их корреляция с семиуровневой моделью OSI, выполняется систематизация и сравнение свойственных им параметров функционирования.

Fieldbus-сети CAN, KNX/EIB, DeviceNet

Controller Area Network (CAN) – стандарт промышленной системы, ориентированный на объединение в единую шину различных исполнительных устройств и датчиков, получивший широкое распространение в автомобильной промышленности, меньшее в промышленной автоматизации и автоматизации управления инженерными подсистемами. Стек протоколов CAN коррелирует с двумя нижними уровнями модели OSI физическим и канальным (рис. 1).

Физический уровень CAN определяет возможные спецификации ISO 11898, ISO 11519-2, SAE J2284, SAE J2411, поддерживающие возможности использования различных сред передачи (оптоволокно, витая пара, радиоканал, ИК-канал, силовая линия), с различными топологиями, методами канального кодирования и, соответственно, скоростями передачи (1000, 500, 125, 10 Кбит/с) технологической ин-

формацией и другими характеристиками (допустимой протяжённостью линии связи, количеством узлов (64) и др.).

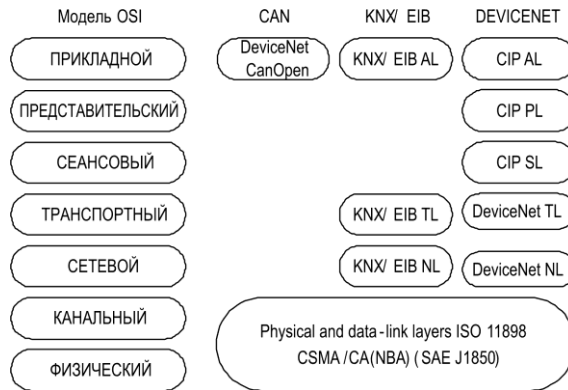


Рис. 1. Стеки протоколов сетей CAN, KNX/EIB, DeviceNet

Информация в CAN передаётся короткими сообщениями-кадрами стандартного формата (кадр данных, удалённой передачи, ошибки передачи, ошибки и повторного запроса передачи). Канальный уровень CAN-сети включает подуровень управления данными (Logical Link Control – LLC), реализующий процедуры обработки ошибки передачи, и подуровень управления доступом к среде передачи (Media Access Control – MAC). Надёжность передачи, определяемая подуровнем LLC, существенно влияет на характеристики передачи информации механизмами обнаружения ошибок (побитовым сравнением отправляемого и принимаемого битов, вставкой бита при последовательной передаче одинаковых битов, проверкой типовых уровней напряжений на шине при передаче), передачей бита подтверждения после успешной передачи, проверки контрольной суммы сообщения.

При корректной (без ошибок передачи) передаче характеристики передачи сети CAN во многом определяются временем доступа и передачи информационных сообщений по каналу промышленной шины согласно алгоритму управления доступа MAC подуровня [1, 2]. В CAN используется алгоритм множественного доступа с контролем несущей (CSMA) и предотвращением (избеганием) коллизий (Collision avoidance) (реже используется обозначение «разрешение коллизий», Collision Resolving – CR) в режиме недеструктивной передачи. Преимуществом, по сравнению с другими алгоритмами, является неразрушающий (не деструктивный, Nondestructive Bitwise Arbitration – NBA) механизм передачи сообщения при коллизии, достигаемый арбитражем

(Arbitration Message Priority – AMP) (управлением) шины по приоритету, что характеризует механизм эффективностью использования канального ресурса промышленной шины. Принцип передачи по указанному протоколу показан в рамках временной диаграммы соперничества узлов за доступ и передачу информации на рисунке 2.

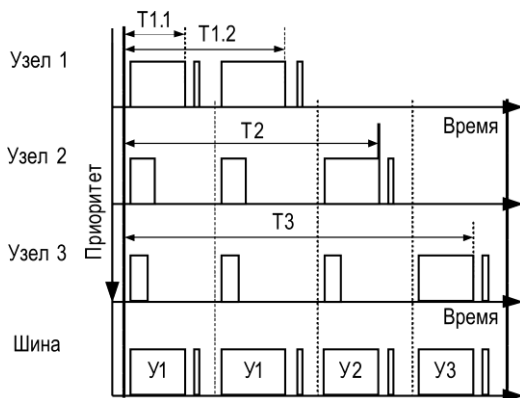


Рис. 2. Пример временной диаграммы взаимодействия узлов по алгоритму CSMA/MA

Согласно методу доступа (рис. 2) узлы дожидаются освобождения канала передачи данных и начинают передачу информационного сообщения, в начале которого содержатся служебные биты приоритета. Узлы, участвующие в передаче, прослушивают канал и в случае установления приоритета больше передаваемого прекращают передачу, переходя в режим прослушивания и приёма данных из канала связи. Появление коллизий в сети возможно в случае практического совпадения моментов времени передачи и приоритетов узлов, что проявляется при загрузенности шины данных выше 30 % [3]. В рабочем режиме коллизии являются редким явлением, что обуславливает детерминированность доступа и передачи сообщения в сети управления. Однако недостатком данного подхода является недетерминированное время передачи сообщений с низким приоритетом при большом количестве узлов в сети и загрузке канала, что обусловлено очередями и коллизиями передачи [3]. При успешном приёме сообщения каждое устройство-приёмник подтверждает получение пакета спустя интервал подтверждения (15 тайм-слотов). При отсутствии подтверждения передача повторяется ещё два раза, после чего передача пакета заканчивается. Для осуществления новой передачи в сети все узлы выжидают межпакетный интервал, равный 50 тайм-слотам, что при скорости передачи,

равной 500 Кбит/с, составляет 100 мкс. Указанный алгоритм и протокольные параметры – временные интервалы и счётчик числа повторных передачи – значительно влияют на характеристики сети, особенно при большой загруженности канала передачи (выше 30 %). Пример коммуникационного взаимодействия по алгоритму доступа показан на рисунке 2, где проиллюстрировано время задержки до успешной передачи узлами с различными приоритетами, в частности большее время доступа и передачи узлом № 3 с минимальным приоритетом.

Базовой спецификации CAN, определённой протоколами канального и физического уровней спецификации CAN Specification 2.0A/B и международного стандарта ISO 11898, часто недостаточно для реализации многих возможностей, требуемых в реальных системах: единообразного управления устройствами различных типов и производителей. В связи с этим вскоре после разработки CAN разрабатываются протоколы прикладного уровня. В число распространённых на данный момент протоколов входят DeviceNet, CANOpen, SDS, CAN Kingdom, SeleCAN, SAE J1939. Прикладные протоколы определили типы прикладных данных и принципы работы с ними, практически не влияющие на уровень вероятностно-временных характеристик передачи и обработки технологической информации в сети.

Отсутствие спецификации сетевого уровня модели OSI делает невозможным организацию распределённой CAN-сети, представляя единую шину с одноадресной передачей данных. Отсутствие протоколов транспортного уровня накладывают ограничение на передачу данных менее 8 байт, автоматического распределения идентификаторов между узлами. Эти предпосылки определяют дальнейшее развитие и применение спецификаций физического и канального уровней стека протоколов CAN в других промышленных системах, в частности KNX/EIB, DeviceNet.

Протокол KNX/EIB (European Installation Bus) представляет собой сеть с собственными протоколами сетевого, транспортного и прикладного уровней OSI поверх физического и канального уровней CAN [4]. Сетевой уровень протокола определяет возможность организации одноадресной и многоадресной (групповой) адресации сообщений. KNX позволяет организовать распределённое взаимодействие, строить иерархии фрагментов шины посредством коммуникационного оборудования – повторителей и маршрутизаторов. Транспортный уровень отвечает за передачу всего объёма данных, мультиплексирование и организацию (фрагментацию) сообщений. В результате введённых дополнений появляется ряд новых факторов, потенциально влияющих на характеристики сети. Производится усложнение алгоритма передачи

подтверждения при многоадресной передаче. Передача подтверждений от нескольких узлов производится одновременно, при этом отрицательный сигнал подтверждения (ввиду ошибки приёма) преобладает над положительным, вследствие чего требуется повторная отправка исходного пакета. Фрагментация данных требует передачи большего числа пакетов данных и времени.

Промышленная сеть DeviceNet используется для связи датчиков, исполнительных механизмов и программируемых логических контроллеров между собой и представляет собой реализацию прикладного, представительского, сеансового, транспортного и сетевого уровней OSI поверх протоколов CAN, дополняя её спецификациями используемых физических сред [4]. Отличительной чертой стека протоколов является протокол реального времени CIP (Common Interface Protocol, информационно-управляющий протокол), покрывший три верхних уровня стека протоколов OSI. Это позволило определить обмен данными через сообщения, а также обмен общими данными в целях конфигурирования, диагностики и управления через явные сообщения. CIP получил широкое распространение и в промышленных сетях, таких как Ethernet/IP, ModbusTCP и др.

Вопросам количественной оценки используемых и составляющих основу моделей рассмотренных сетей посвящены работы [1, 2, 4-6].

Fieldbus-сети Ethernet/IP, ModbusTCP

Ethernet/IP (Industrial Ethernet) – вариант технологии Ethernet для применения в промышленности для организации обмена данными между программируемыми контроллерами и системами человеко-машинного интерфейса, подключения к контроллерам удалённого обслуживания (датчиков и исполнительных устройств). Широко применяется при автоматизации инженерных подсистем зданий.

Ethernet/IP представляет реализацию протокола CIP (верхних трёх уровней OSI) над стеками протоколов TCP/IP (транспортного и сетевого уровней) и Ethernet (IEEE 802.3) (канального и физического) (рис.3).

Широкому применению Ethernet для решения промышленных задач реального времени препятствует суть метода управления случайным доступом узлов к каналу шины данных. Метод множественного доступа CSMA с обнаружением коллизий в канале (Collision Detection – CD) не гарантирует обмен информацией между датчиками и исполнительными механизмами в реальном времени, что обусловлено происходящими коллизиями и характерно в основном для шинной и свободной топологий сети.

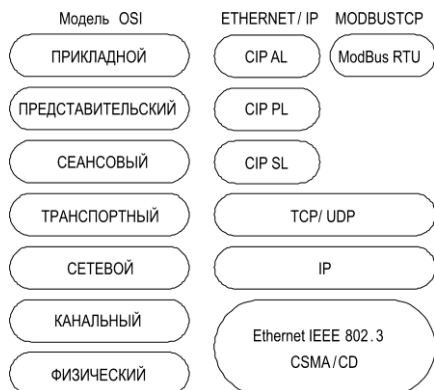


Рис. 3. Стеки протоколов сетей Ethernet/IP и ModbusTCP

Алгоритм передачи согласно протоколу множественного доступа с обнаружением коллизий реализуется следующим образом [7]. Каждый узел сети может передавать данные в любой момент времени (протокол 1-Persistent – узел начинает передачу пакета с вероятностью 1,0, как только определяет, что канал свободен), определяя асинхронные пакетные циклы передачи (синхронность разделения канала на циклы передачи возникает при организации очереди сообщений для передачи несколькими узлами).

Передаче пакета предшествует 96-битный анализ занятости среды передачи (межпакетный интервал), и, если она свободна, начинается передача пакета. Поскольку первый бит одного передаваемого пакета может достигнуть остальных с некоторой задержкой, то может возникнуть псевдоодновременная передача, результатом которой является коллизия. Если передаваемый и принимаемый сигналы из шины отличаются, то узел фиксирует факт обнаружения коллизии и для увеличения вероятности обнаружения коллизии остальными узлами сети прерывает передачу и «усиливает коллизию» посылкой в сеть специальной 32-битовой jam-последовательности.

Повторная попытка отправки пакетов, включая обычный межпакетный интервал, осуществляется спустя случайное количество тайм-слотов ожидания продолжительностью 512 бит каждый, выбранное равномерно из диапазона $[0 \dots 2N \leq \text{backoffLimit} - 1]$, где backoffLimit – максимальное число возрастных диапазонов паузы повторения передачи, N – номер попытки отправки пакета, и при условии, что канал свободен.

Предельное число попыток передачи составляет 16, если число попыток исчерпано, то передаваемый кадр удаляется. Пример комму-

никационного взаимодействия (с транспортным протоколом UDP) трёх узлов с общим количеством пакетов передачи 4, коллизией передачи и повторной отправкой иллюстрирует время сериализации пакетов в сети (рис. 4). На временной диаграмме проиллюстрированы моменты псевдоодновременных передач сообщений узлов 1–3, приводящих к коллизиям, откладывающим и увеличивающим время успешной передачи сообщений других узлов.

Предусмотренный стандартом Ethernet IEEE 802.3 механизм разрешения коллизий данных порождает нерегулярные задержки при передаче пакетов, в отличие от CSMA/NBA. Высокая скорость передачи данных (10 Мбит/с для сетей датчиков и исполнительных механизмов) не может гарантировать время передачи с высокой вероятностью, особенно при загрузке канала выше 20 %, поэтому для достижения быстрого действия реального времени промышленный Ethernet применяет различные меры для избегания коллизий. Кардинальным подходом является построение топологий промышленной шины типа «звезда», в центре которых располагается высокопроизводительное коммуникационное оборудование (маршрутизаторы), реализующие арбитраж и дисциплину приоритетного построения и обслуживания очереди сообщений, поступающих с дуплексных линий связи узлов.

Преимуществом похода является практическое устранение коллизий. Недостатками являются: увеличение стоимости сети за счёт необходимого высокопроизводительного оборудования (маршрутизаторов) и протяжённости кабельной системы; увеличение задержки, связанное с необходимостью передачи через несколько каналов, что особо характерно при низких нагрузках каналов, и образование очереди передачи при большой нагрузке.

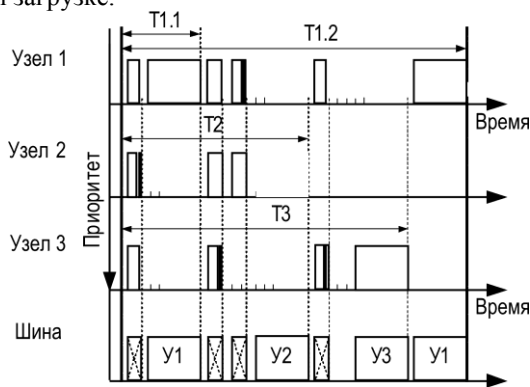


Рис. 4. Пример временной диаграммы взаимодействия узлов по алгоритму CSMA/CD

Сетевой уровень OSI в рамках стека fieldbus-сети обеспечивает обмен сообщениями между различными сегментами шины данных, обеспечивая адресацию согласно распространённому протоколу IP (Internet Protocol). Протокол предусматривает возможность организации одноадресной и многоадресной передачи. Транспортный уровень обеспечивает передачу данных с заданным уровнем надёжности доставки информации, реализуемой протоколами TCP и UDP. Методы обеспечения надёжности данного уровня являются значимым фактором, влияющим на производительность.

Протокол передачи UDP (User Datagram Protocol, пользовательский дейтаграммный протокол) не определяет необходимость дополнительного ряда передач, и поэтому используется в случаях, когда первоочередным фактором становится время доставки, а не надёжность передачи данных. TCP (Transmission Control Protocol, протокол управления передачей) представляет собой протокол со средствами установки, подтверждения и завершения соединения, со средствами обнаружения и коррекции ошибок.

Высокая надёжность передачи данных достигается ценой дополнительных временных задержек и увеличения объёма передаваемой информации. Сегменты данных, которые передаются и должны быть подтверждены, могут быть «утрачены» при доставке и будут требовать повторной передачи, что также определяет увеличение времени доставки сообщений и нагрузки на канал промышленной шины. В таком случае характеристики сети становятся ещё менее детерминированными, а её анализ очень сложным, и поэтому TCP используется для передачи некритичных ко времени данных (конфигурации, загрузка/выгрузка программ) при очень малых нагрузках канала (<15 %) [3, 7].

Протоколы и алгоритмы вышележащих уровней сети Ethernet/IP, коррелирующие с уровнями 5–7 стека OSI, отвечают за конечную интерпретацию передаваемой технологической информации и практически не влияют на характеристики передачи.

Аналогом промышленного Ethernet с позиции использования стека TCP/IP и нижних уровней Ethernet является большое количество промышленных сетей EtherCat, ProfiNet, PowerLine и другие, различия которых заключаются в степени модификаций стандартного стека TCP/IP: функции синхронизации, новые сервисы информационного обмена (доставки сообщений), диагностические функции, методы самокорректировки. Канальный и физический уровни Ethernet при этом остаются неизменными. Промышленный Ethernet поддерживает возможность взаимодействия (путём шлюзования) с другими сетями по-

средством шлюзования с ControlNet, DH+, ProfiNet, DeviceNet, ModbusTCP и др.

ModbusTCP является версией промышленной шины реального времени Modbus с её прикладным уровнем, но инкапсуляцией данных в пакеты стеков протоколов TCP/IP и Ethernet (IEEE 802.3). Указанное определяет отличие ModbusTCP от Ethernet/IP только на прикладном уровне. Отличительной чертой сетевого уровня протокола от Ethernet/IP является отсутствие поддержки широковещательной и многоадресной адресации, а поддержки только одноадресной коммуникации. ModbusTCP поддерживает коммуникацию с Ethernet/IP, ProfiNet и рядом других сетей.

Проблеме разработки математических моделей fieldbus-сетей, в основе которых лежат методы множественного случайного доступа узлов CSMA/CD, посвящено значительное количество работ [8, 9–11, 12, 13].

Fieldbus-сети BacNet, LonWorks

BacNet – промышленная сеть с одноимённым протоколом, применяемая в сетях автоматизации зданий и управления технологическими процессами. На физическом и канальном уровнях модели OSI поддерживает стек протоколов Ethernet, при этом сетевой и прикладной уровни определены собственной спецификацией BacNet (в BacNet/IP используются протоколы UDP/IP [14]) и определяют, соответственно, сервисы адресации, классы и сервисы прикладных задач (тревоги и события, управление устройством и др.) (рис. 5).

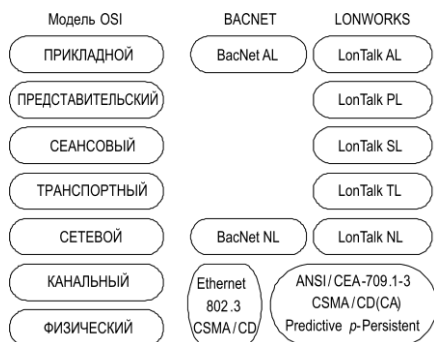


Рис. 5. Стеки протоколов сетей BacNet и LonWorks

Широкое распространение в промышленной шине BacNet получают протоколы доступа и передачи физического и канального уров-

ней, способствующие более эффективному [14] использованию полосы пропускания каналов, обеспечиваемому синхронным прогнозированием нагрузки узлов на канал и избеганием (предсказанием) коллизий. Данный протокол впервые разработан для сети автоматизации LonWorks и носит название Predictive p-Persistent CSMA (PpP-CSMA) (прогнозирующий с псевдопостоянной шириной временного окна доступа (псевдо-постоянной настойчивости)). Для обозначения данного протокола применяется также обозначения «с обнаружением коллизий» (CD), «с избеганием коллизий» (CA) путём приспособления узлов к соперничеству за доступ к каналу. Узел соперничает за совместно используемый канал согласно р-CSMA-протоколу, успешно начинает передачу сообщения с вероятностью p , если канал неактивен, и задерживает передачу с вероятностью $(1 - p)$ [15].

Оптимальность использования канала в р-CSMA сильно зависит от значения p , которое представляет уровень настойчивости протокола. В частности, большие значения p вызывают чрезмерные коллизии в сети, в то время как маленькие значения вероятности передачи ухудшают использование пропускной способности, приводя к неактивности канала. Таким образом, необходим компромисс между большими и маленькими значениями p , чтобы обеспечить использование пропускной способности на удовлетворительном уровне. Однако постоянный уровень настойчивости, максимизирует пропускную способность только для предварительно выбранного числа узлов, конкурирующих за канал, что значительно ограничивает полезность чистого р-CSMA на практике, в том числе 1-CSMA/CD Ethernet, используемого в распределённых системах управления Ethernet/IP, ModbusTCP и других шинах Industrial Ethernet.

В случае, когда количество узлов, готовых к передаче, неизвестно или изменяется во времени, значение p не может быть установлено оптимально, и, следовательно, производительность р-CSMA может быть значительно уменьшена. В связи с этим для решения задач выбора оптимального уровня настойчивости в Lon-сети используется прогнозирующий (predictive) р-CSMA-протокол с предотвращением коллизий, что позволяет более эффективно использовать пропускную способность на большом диапазоне загрузки сетевого канала (до 60 %).

Predictive p-Persistent CSMA протокол является адаптивной версией чистого р-CSMA и разработан специально для локальных операционных сетей (Lon), объединяющих интеллектуальные датчики и исполнительные механизмы. Вероятность успешной передачи p становится переменной величиной и динамически корректируется со-

гласно ожидаемой нагрузке, уменьшается в случае коллизий и увеличивается после каждой успешной передачи [15].

Передача данных в Lon-сети выполняется полностью синхронными пакетными циклами (packet cycle). Циклы принято разделять на свободные, когда в канале нет активности после передачи, и занятые, когда выполняется передача. Каждый занятый пакетный цикл представляет собой время передачи пакета с сообщением и время задержки, необходимое для того, чтобы узлы в сети смогли начать передачу. Алгоритм доступа к среде CSMA включает следующие этапы [14, 16]. Каждый пакетный цикл начинается с межпакетного интервала (68–72 тайм-слотов) для определения того, что канал свободен. После следует интервал времени (используется только после занятого пакетного цикла), равный фиксированному для узла количеству приоритетных слотов доступа, каждый продолжительностью 13 тайм-слотов. После приоритетного воспроизводится случайных интервал доступа для избегания коллизий. Узел формирует случайную задержку T , которая определяется случайным количеством тайм-слотов продолжительностью 13 бит каждый. Если канал после случайной задержки T свободен, то узел начинает передачу пакета. В противном случае узел ждёт освобождение канала и снова пытается конкурировать за канал (начиная с 1-го этапа). В случае, если несколько узлов выбрали одинаковое количество временных слотов доступа и начали передачу одновременно, в сети возникает коллизия. Все пакеты, участвующие в столкновении, становятся поврежденными и требуют повторной передачи. Традиционно для LON-сетей передача при коллизии может быть остановлена, если обнаружена в момент времени, не превышающий 25% преамбулы пакета, однако наиболее вероятна ситуация без остановки передачи, требующая ожидания завершения передачи в пакетном цикле. Случайное время задержки T определяется как псевдослучайное число тайм-слотов, взятых согласно равномерному распределению из интервала $[0 \dots (BL \cdot W_{\text{base}} - 1)]$, где W_{base} (W) – ширина основного соревновательного окна, равная 16 слотам для EIA-709; BL – величина отставания канала (backlog) (от намеченной работы), которая характеризует загруженность канала, изменяется от 1 до 63. Для свободного канала BL равняется единице, и соревновательное окно W содержит W_{base} временных слотов. Когда нагрузка на канал увеличивается, количество слотов растёт, так как растёт отставание канала BL , и ширина соревновательного окна может достигать максимального значения $W = 63 \cdot W_{\text{base}} = 1008$ слотов. Таким образом, уровень настойчивости р-CSMA равняется $1/(BL \cdot W_{\text{base}})$ и является переменным, имеет нижнюю границу вероятности успешной передачи одним узлом

($1/16 = 0,0625$) и верхнюю границу ($1/1008 = 0,0009$). Пример коммуникационного взаимодействия (с транспортным сервисом с подтверждением узла 1) трёх узлов с общим количеством сообщений передачи, равным 4, коллизией передачи и повторной отправкой иллюстрирует время сериализации сообщений различных узлов (рис. 6).

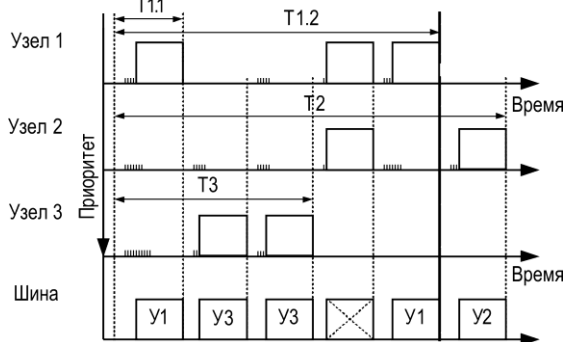


Рис. 6. Пример временной диаграммы взаимодействия узлов по алгоритму PpP-CSMA

Сложность настоящего протокола доступа, по сравнению с протоколами CSMA/NBA и 1-CSMA/CD, заключается в высокой зависимости алгоритма прогнозирования нагрузки Backlog от факторов функционирования вышележащих сетевого, транспортного и сеансового уровней модели OSI, определяющих типы информационного трафика: сервисы доставки и виды адресации, таймеры передачи. Кроме этого, процесс передачи ограничен большим количеством счётчиков и таймеров передачи (таймер ожидания, подтверждения, передачи и др.), что значительно влияет на характеристики информационного взаимодействия элементов сети управления. Указанное большое число факторов функционирования делает протокол PpP-CSMA сложным для количественного анализа характеристик по сравнению с CSMA/NBA, CSMA/CD. Это определяет малую проработанность моделей промышленных сетей с протоколом данного типа и низкую корректность результатов оценки характеристик в известных публикациях [14, 15, 17-19]. Однако протокол характеризуется более эффективностью, чем CSMA/CD, использованием полосы пропускания канала на широком диапазоне загрузки канала, и является актуальным для исследования.

Заключение

Обзор коммуникационных стеков протоколов распространённых fieldbus-сетей выявил общность свойственных им алгоритмов и особенностей функционирования по уровням модели OSI. Общность алгоритмов доступа узлов к каналу передачи, основанных на протоколах множественного доступа с контролем несущей (CSMA) (физического и канального уровней OSI), заключается в общих принципах организации доступа и передачи с особенностями межпакетных временных интервалов доступа, продолжительности передачи сообщений. Отличительными чертами являются механизмы разрешения коллизий и синхронность пакетных циклов передачи.

Обзор работ показывает схожие аналитические и имитационные методы построения и расчёта характеристик моделей сетей. Указанное, наряду с недостаточной детализацией существующих моделей сетей и низкой адекватностью оценки их характеристик, определяет целесообразность проработки методики обобщённого анализа систем представленного класса. В связи с этим для получения обобщённых результатов, для дальнейшего исследования, создания модели сети и способа оценки её характеристик выбран наиболее сложный и малоисследованный семиуровневый стек протоколов LonTalk сети LonWorks.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта №18-37-00070 "Алгоритмы проектирования информационно-управляющих сетей LonWorks с требуемыми вероятностными и временными характеристиками".

Библиографический список

1. *Lian, F.-L.* Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet. Technical Report: UM-MEAM-99-02 / *F.-L. Lian, J.R. Moyne, D.M. Tilbury.* – 1999. – Режим доступа: <http://www-personal.umich.edu/~tilbury/papers/lmt99csm.pdf> (дата обращения: 28.09.2016).
2. *Tindell, K.* Calculating controller area network (CAN) message response times / *K. Tindell, A. Burns, A.J. Wellings* // *Control Engineering Practice.* – 1995. – Vol. 3, iss. 8. – P. 1163–1169.
3. *Новиков, Ю.В.* Локальные сети: архитектура, алгоритмы, проектирование / *Ю.В. Новиков, С.В. Кондратенко.* – М.: ЭКОМ, 2000. – 308 с.
4. *Lian, F.-L.* Network design consideration for distributed control systems / *F.-L. Lian, J. Moyne, D. Tilbury* // *IEEE Transactions on*

- Control Systems Technology. – 2010. – Vol. 10, iss. 2. – P. 297-307.
5. *Hong, S.H.* Bandwidth allocation scheme in the CAN protocol / *S.H. Hong, W.H. Kim* // Control Theory and Applications: IEEE Proc. – 2000. – Vol. 147. – P. 37–44.
 6. *Hong, S.-H.* A bandwidth allocation scheme in fieldbuses / *S.-H. Hong, J.-H. Lee* // International Journal of Control, Automation, and Systems. – 2010. – № 8 (4). – P. 831–840.
 7. *Олифер, В.Г.* Компьютерные сети. Принципы, технологии, протоколы: учеб. для вузов / *В.Г. Олифер, Н.А. Олифер.* – 4-е изд. – СПб.: Питер, 2010. – 944 с.
 8. *Бочаров, П.П.* Теория массового обслуживания / *П.П. Бочаров, А.В. Печинкин.* – М.: Изд-во РУДН, 1995. – 529 с.
 9. *Назаров, А.А.* Исследование сети связи, управляемой адаптивным протоколом случайного множественного доступа, в условиях критической загрузки / *А.А. Назаров, Д.Ю. Кузнецов* // Проблемы передачи информации. – 2004. – № 3. – С. 69–80.
 10. *Назаров, А.А.* Математическое моделирование сети связи случайного множественного доступа / *А.А. Назаров, А.Н. Туенбаева* // Информационные технологии и математическое моделирование: материалы III Всерос. науч.-практ. конф. – Томск, 2004. – С. 51–56.
 11. *Назаров, А.А.* Исследование компьютерной сети связи, управляемой протоколом случайного множественного доступа / *А.А. Назаров, А.Н. Туенбаева* // Вестник Новосибир. гос. ун-та. Информационные технологии. – 2005. – Т. 2, вып. 1. – С. 74–80.
 12. *Falin, G.I.* A survey of retrial queues / *G.I. Falin* // Queueing Systems. – 1990. – Vol. 7. – P. 127–167.
 13. *Falin, G.I.* On sufficient conditions for ergodicity of multichannel queueing systems with repeated calls / *G.I. Falin* // Advanced in Applied Probability. – 1984. – Vol. 16. – P. 447–448.
 14. *Дитрих, Д.* LON-технология, построение распределенных приложений / *Д. Дитрих, Д. Лой, Г.Ю. Швайцгер;* пер. с нем. под ред. *О.Б. Нuzамутдинова.* – Пермь: Звезда, 1999. – 242 с.
 15. *Miśkiewicz, M.* Analysis of mean access delay in variable-window CSMA / *M. Miśkiewicz* // Sensors. Schweiz: Molecular Diversity Preservation International. – 2007. – Vol. 7. – P. 3535–3559.

16. LonTalk protocol specification: ANSI/CEA-709.1-B. – United States, 2006.
17. *Buchholz, P.* Analytical analysis of access-schemes of the CSMA type / *P. Buchholz, J. Plonnigs* // Proc. of IEEE International Workshop on Factory Communication Systems WFCS. – 2004. – Wien, 2004. – P. 127–136.
18. *Miśkowicz, M.* Latency characteristics of event-driven task scheduler embedded in neuron chip / *M. Miśkowicz* // IJCSNS International Journal of Computer Science and Network Security. – 2007. – Vol. 7, iss. 12. – P. 132–149.
19. *Moshe, K.* Collision resolution simulation for distributed control architectures using LonWorks / *K. Moshe* // IEEE International Conference on Automation Science and Engineering. – Edmonton, 2005. – P. 319–326.

В.В. Макаров, В.В. Ланин ³

Национальный исследовательский университет
«Высшая школа экономики»
(Пермский филиал)
vladislavmakarov1998@gmail.com, vlanin@hse.ru

ПРОФИЛИРОВАНИЕ GATE DEVELOPER ДЛЯ ВЫЯВЛЕНИЯ ПРИЧИНЫ ПЕРЕПОЛНЕНИЯ ПАМЯТИ

Статья подготовлена в ходе проведения исследования по проекту № 17-05-0020 в рамках Программы Научного фонда Национального исследовательского университета «Высшая школа экономики» (НИУ ВШЭ) в 2017-2018 гг. и в рамках государственной поддержки ведущих университетов Российской Федерации «5-100». В статье исследуется проблема переполнения памяти в приложении для лингвистического анализа GATE Developer, перечисляются причины появления проблемы, выбирается подходящий инструментарий для проведения профилирования, проводится профилирование с помощью средства JProfiler с выявлением закономерностей и даются варианты решений.

Ключевые слова: профилирование, переполнение памяти, лингвистический анализ.

Введение

Научно-учебная группа «Разработка программного обеспечения для проведения корпусных исследований английского языка» НИУ ВШЭ г. основной целью которой является разработка программного обеспечения для проведения исследований стиля письменной научной речи на английском языке. Ключевые атрибуты (лексические, грамматические, синтаксические и др. языковые явления), научного стиля речи распределяются в текстах научных статей с определённой закономерностью и на больших корпусах эти закономерности могут быть выявлены с помощью разрабатываемого программного обеспечения. Планируется создание исследовательского инструмента для количественного анализа научной речи и обучающей программы, облегчаю-

³ Работа выполнена при поддержке Научного фонда НИУ ВШЭ (проект № 17-05-0020) в рамках государственной поддержки ведущих университетов Российской Федерации «5-100»

щей преподавание английского языка для академических целей. Создаваемое ПО также позволит оптимизировать процесс обучения и оценки успешности прохождения дисциплины «Академическое письмо на английском языке». Полученные эмпирические данные могут быть применены в педагогических и исследовательских целях.

Одним из программных средств, используемых для анализа работ, является приложение GATE Developer. Данное средство написано на языке Java и широко используется в лингвистическом анализе, поскольку в него входят разнообразные модули, позволяющие создавать корпуса, выбирать и настраивать методы, с помощью которых они будут исследоваться. Ещё одним плюсом является доступность одноимённой библиотеки (Java), позволяющий самостоятельно разрабатывать свои продукты и создавать расширения. Среди расширений, которые используются исследовательской группой, являются созданные в рамках ВКР расширения [1].

Проблема переполнения памяти

Во время работы упомянутой программы возникла проблема, связанная с прекращением обработки подкорпуса британского национального корпуса (baby BNC), предоставленного группе в исследовательских целях. Возникающая в определённый момент ошибка «Exception in thread “AWT -EventQueue-0” java.lang.OutOfMemoryError: Java heap space» связана с переполнением памяти, выделяемой GATE Developer, а именно виртуальной машине Java, на которой работает приложение Gate Developer.

Для того чтобы понять, в какой конкретной ситуации начинает использоваться больше памяти, чем необходимо, нужно проследить, что влияет на увеличение затрачиваемой памяти. Проанализировав функциональность GATE Developer, можно спрогнозировать результат проверки. Мы предполагаем, что факторами, негативно влияющими на оперативную память, занимаемую приложением, являются:

- размер и количество файлов, участвующих в работе отдельного проекта.
- некоторые объекты не удаляются из памяти, другими словами, проблема в сборщике мусора.

Выбор инструментария

Поскольку GATE Developer и написанные для неё плагины были созданы на языке Java, необходимо найти качественный профилировщик. На рынке ПО имеется множество продуктов, позволяющих про-

вести профилирование приложений, созданных на Java. Среди них – JProfiler [2], YourKit Java Profiler [3].

Несмотря на то, что перечисленные продукты являются платными, пользователям предоставляется trial-версия, чтобы оценить их функциональность. Эти продукты имеют схожие функции:

- присутствует возможность подключаться к уже работающему приложению на локальной машине и следить за ним во время его работы;
- реализована функция сохранения снимков – записанной работы приложения;
- позволяет отслеживать следующие данные: использование памяти, список использованных объектов, количество потоков, загруженность процессора и т.д.

JProfiler обладает меньшим периодом временного использования – 10 дней, однако потребителю позволяет очищать информацию о лицензии и получать новый ключ продукта, что ведёт к возможности длительного использования без потери функциональности.

Сценарий и результат тестирования в JProfiler

Для обработки были выбраны:

- Британский корпус из 50 различных текстов, тип хранения – хранилище данных.
- Корпус из 13 выбранных и отформатированных статей категории Computer Science, статьи в формате текстовых документов.
- Корпус с большим количеством документов.

Плагины, с помощью которых должен проводиться анализ, выбраны в соответствии со списком расширений, которые используются исследовательской группой, о чем упомянуто выше.

Затем были выбраны действия, которые следует выполнять при профилировании, и параметры. В данном случае их три: выделяемая память отвечает за количество памяти, даваемой виртуальной машине Java; используемая память отвечает за то, сколько памяти используется объектами и пакетами; пик использования памяти позволяет проследить, в какой момент увеличивается выделяемая память. Затем было проведено эксперименты (см. табл. 1).

Действительно, при достижении определённого параметра памяти возникает упомянутое исключение о её переполнении. В профилировщике видно, что максимальный размер памяти, доступный GATE, равен 1,23 ГБ, и при попытке увеличить количество выделяемой памяти выходит исключение.

Таблица 1. Список экспериментов

Действие	Выделяемая память	Используемая память	Пик использования памяти
Добавление плагинов	204 МБ	39 МБ	161 МБ
Добавление обрабатываемых ресурсов	204 МБ	111 МБ	171 МБ
Добавление хранилища данных и британского корпуса	204 МБ	140 МБ	141 МБ
Создание обрабатывающего приложения и обработка британского корпуса	412 МБ	216 МБ	331 МБ
Добавление 13 статей	470 МБ	245 МБ	386 МБ
Создание корпуса из 13 статей	470 МБ	353 МБ	353 МБ
Обработка корпуса из 13 статей	596 МБ	343 МБ	525 МБ
Повторная обработка корпусов: 4 цикла	669 МБ	355 МБ	592 МБ
Создание корпуса из 130 документов	1,23 ГБ	0,6 ГБ	1,16 ГБ

Как можно заметить (рис. 1), выделение дополнительной памяти приложению происходит, когда используемая память достигает какой-то границы. Стоит уточнить, что увеличение памяти происходит не мгновенно до нового значения, а после 1-3 скачков, в зависимости от разницы между используемой памятью и выделяемой. Если приложение будет требовать больше памяти, то увеличение выделяемой произойдёт. Поэтому пиковые значения предшествовали увеличению.

Также из данных (рис. 2) можно получить вывод, что наибольший вред памяти нанесло создание корпуса из большого числа документов, а добавление расширений или создание меньшего по размеру корпуса не так сильно повлияло на память. Таким образом, подтвердилась гипотеза, что именно количество и размер обрабатываемых документов влияет на размер используемой памяти.

Решение проблемы

Для того, чтобы решить проблему переполнения памяти, можно воспользоваться следующими способами:

- Каждый раз нужно перезапускать GATE Developer. Есть два минуса: может занять много времени; проблема всё равно может возникнуть при создании приложения с большим количеством текста, приложений и плагинов.

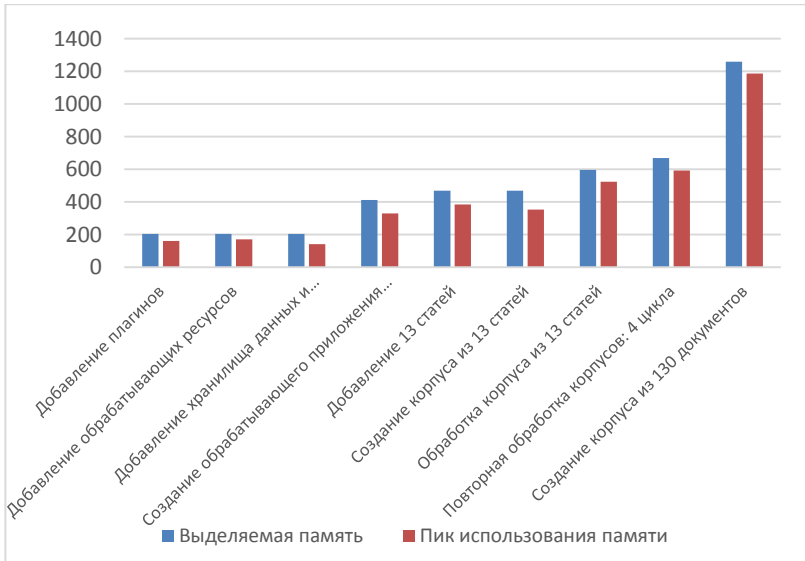


Рисунок 1. Использование памяти

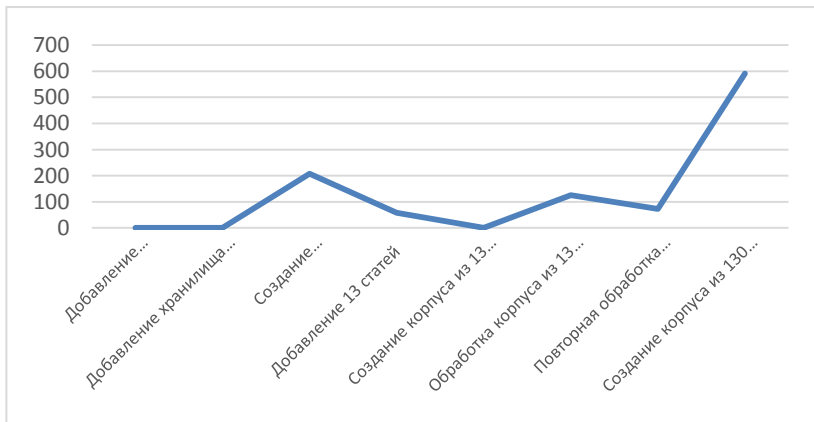


Рисунок 2. Разница в выделяемой памяти

- Можно увеличить количество выделяемой памяти для GATE Developer через конфигурацию Java на локальной машине. Выполняется это путём добавления параметров `-Xms1024m` (выделение на момент старта приложения, 1gb, например) и -

Xmx4096m (максимально доступное количество памяти для приложения, 4gb). В конкретном случае параметры добавляются в файл конфигурации gate.l4j.ini в папке с приложением [4]. Здесь тоже возникают минусы: для каких-то больших проектов на локальной машине может быть недостаточно памяти, и проблема может возникнуть повторно.

- Создать отдельное приложение со схожим функционалом, как и у GATE Developer, но с тем условием, что оптимизация памяти будет на уровне выше, чем в упомянутом приложении. Минус этого решения – трудозатраты на создание отдельного приложения.

Заключение

В ходе исследования причин переполнения памяти в GATE Developer было проведено профилирование с помощью JProfiler, которое показало, что наибольший вес в использовании памяти вкладывает количество и размер обрабатываемых документов. Затем были предложены варианты решения, как минимизировать появление этой проблемы. Также стоит упомянуть, что проблема переполнения памяти появляется не только в исследуемом приложении, но и в других, созданных на Java. Множество продуктов с качественной архитектурой и программным кодом сталкиваются с ней, когда требуется загружать дополнительные ресурсы в приложения. Параметры виртуальной машины Java, конечно, могут исправить ситуацию, но требуется производственная машина для использования этих параметров.

Список литературы

1. *Шучалова Ю.С.* Автоматизация проверки соответствия текстов учебных работ студентов академическому стилю английского языка // С. 32-45.
2. The definitive guide to JProfiler [Электронный ресурс]. – URL: <https://www.ej-technologies.com/resources/jprofiler/help/doc/JProfiler.pdf> (дата обращения - 20.11.2018).
3. YourKit Java Profiler Knowledge Base [Электронный ресурс]. – URL: <https://www.yourkit.com/docs/> (дата обращения - 20.11.2018).
4. GATE Documentation [Электронный ресурс]. – URL: <https://gate.ac.uk/documentation.html> (дата обращения - 2.11.2018).

В.Д. Марквирер, А.О. Суворов

Национальный исследовательский университет
«Высшая школа экономики»
(Пермский филиал)
Vladlena_Markvirer@mail.ru

Пермский национальный исследовательский
политехнический университет
AOSuvorov@pstu.ru

ПАТТЕРНЫ, ТЕХНОЛОГИИ И ИНСТРУМЕНТЫ МОБИЛЬНОЙ РАЗРАБОТКИ ПРИКЛАДНЫХ ПРОГРАММ ДЛЯ СПОРТА

В статье приведён анализ паттернов разработки программного обеспечения, инструментов и сред создания мобильных приложений, технологии реализации аналитики данных, хранимых в приложении. Практическая часть работы представлена кратким описанием создания мобильного приложения для велосипедиста по выбранным методикам, подтверждающим возможность реализации поставленных заказчиком задач с помощью описанных технологий.

Ключевые слова: MVC, Android, Android Studio, Java, мобильное приложение, спорт.

Введение

Мобильные приложения в современном мире играют одну из основных ролей в жизни большинства людей. Для многих использование смартфонов, планшетов является фактором успешности и самоорганизованности, потому что они всегда находятся поблизости, можно в любой момент сделать заметку, отметить событие в календаре, внести изменения в какой-либо документ и т.п. Существуют также специальные приложения, которыми удобно пользоваться для занятий спортом, ведения журнала тренировок, контроля своего здоровья и др. К сожалению, не все приложения доступные в магазинах на различных платформах (например, Play Market, App Store) являются хорошо разработанными, предоставляют именно те функции, которые необходимы пользователям, позволяют настраивать интерфейс «под себя». Зача-

стую готовые решения можно скачать бесплатно, где основная функциональность присутствует, однако, чтобы воспользоваться уникальными возможностями той или иной программы, нужно купить это приложение, иногда за достаточно высокую цену. Поэтому создаётся впечатление однообразности приложений, нет возможности выбора.

Чтобы исправить ситуацию, было принято решение реализовать такое мобильное приложение, в котором спортсмены, в частности, велосипедисты могли бы вести учёт своего тренировочного процесса, контролируя свои показатели, используя статистический модуль, сравнивая аналогичные периоды разных сезонов и корректируя свой тренировочный процесс на основе полученных данных.

Для реализации поставленной цели необходимо понять, как вообще создаются мобильные приложения, какие модели, подходы, паттерны используются, в какой среде можно написать полнофункциональное спортивное мобильное приложение «с нуля», а также какие технологии и библиотеки будут полезны при разработке.

Паттерны, технологии, инструменты

Выбор паттерна разработки

Паттерны используются разработчиками как мобильных, так и других приложений, для разделения функций взаимодействия пользователя с интерфейсом и данными, т.е. разделение интерфейса и логики. В статье будут кратко рассмотрены три паттерна: *MVC* (см. рис. 1), *MVP* (см. рис. 2) и *MVVM* (см. рис. 3).

MVC (Model-View-Controller) – универсальная программная архитектура для делегирования функций приложения на три модуля [1]: *Модель (Model)* – классы, объекты, атрибуты, база данных, *Вид/Представление (View)* – макеты, окна, кнопки, текстовые поля, т.е. всё то, с чем может взаимодействовать пользователь напрямую, *Контроллер (Controller)* – бизнес-логика приложения, т.е. описание реакции на действие пользователя [2]. Паттерн MVC является гибким, потому что может расширяться на серверные и клиентские приложения [3].

MVP (Model-View-Presenter) – паттерн, аналогичный MVC, но контроллер заменён на *презентёра (Presenter)*, который имплементирует интерфейс представления (получает ссылку на него, обычно в конструкторе), реагирует на события представления и меняет модель по полученному запросу. Интерфейс представления имеет набор свойств и методов. Каждое представление должно реализовывать соответствующий интерфейс, а логика представления должна иметь ссылку на экземпляр презентёра [2].

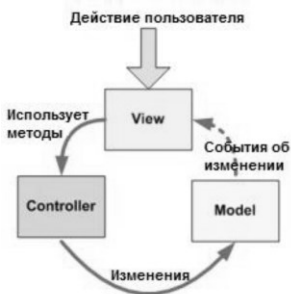


Рис. 1. Паттерн MVC

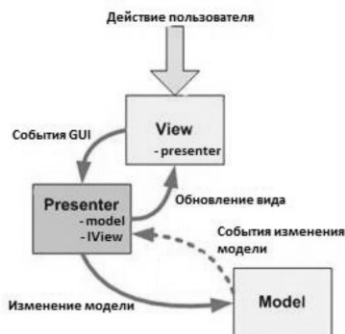


Рис. 2. Паттерн MVP

MVVM (Model-View-View_Model) – паттерн, в котором подразумевается абстрагирование слоёв архитектуры друг от друга. Представление не реализует интерфейс, но имеет ссылку на источник данных. Элементы представления связаны со свойствами и событиями соответствующей *модели представления (View_Model)*, которая должна реализовывать интерфейс для автоматического обновления элементов представления [2].

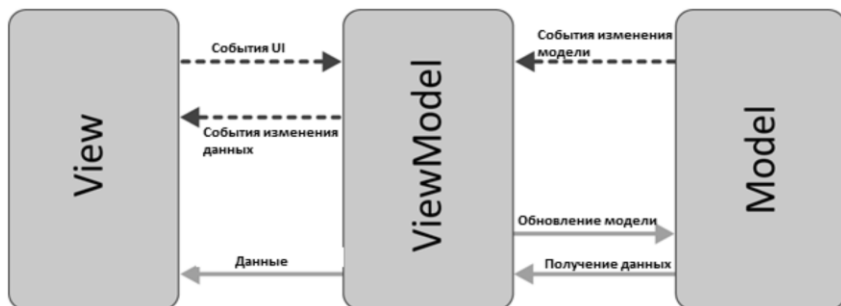


Рис. 3. Паттерн MVVM

Ниже представлена сравнительная таблица (см. табл. 1) рассмотренных паттернов разработки, после анализа которой принято решение, что наиболее подходящим паттерном для разработки мобильного приложения является MVC, потому что есть чёткое разделение функций элементов архитектуры, MVC более гибкий для практического применения, а также на основании рассмотренных для сравнения критериев.

Таблица 1. Сравнительная таблица паттернов архитектуры

Критерий	MVC	MVP	MVVM
<i>Основная идея</i>	Контроллер и представление зависят от модели, модель не зависит от двух компонент	Создание абстракции представления за счёт создания интерфейса.	Связывание элементов представления со свойствами и событиями View-модели.
<i>Использование</i>	Связь между представлениями и другими частями приложения невозможна	Невозможно связывание данных (Binding).	Возможно связывание данных без необходимости ввода специальных интерфейсов представления (отсутствует необходимость реализации IView).
<i>Связывание представления с моделью</i>	Больше возможностей по управлению представлением.	Необходимо программировать.	Автоматически.
<i>Эл-т различия</i>	Контроллер	Презентёр	View-модель
<i>Признаки эл-та различия</i>	<ul style="list-style-type: none"> - определяет, какое представление должно отображаться в текущий момент; - события влияют только на контроллер, контроллер влияет на модель и меняет представление; - несколько представлений одного контроллера. 	<ul style="list-style-type: none"> - двустороннее взаимодействие с представлением; - Представление взаимодействует прямо с презентёром путём вызова соответствующих функций или событий экземпляра презентёра; - взаимодействие с View происходит за счёт реализации представления специальным интерфейсом; - один экземпляр презентёра связан с одним отображением. 	<ul style="list-style-type: none"> - двустороннее взаимодействие с представлением; - View-модель – абстракция представления (свойства представления – свойства абстракции); - нет ссылки на интерфейс представления, т.е. изменение абстракции меняет представление за счёт механизма связывания данных (Binding); - один экземпляр View-модели – одно отображение.
<i>Степень изученности</i>	Есть опыт работы	Теоретический опыт	Теоретический опыт
<i>Пример использования</i>	ASP.NET MVC	Windows Forms	WPF

Выбор среды разработки

Следующим шагом является выбор среды *IDE* или интегрированной среды разработки. Многие эксперты настаивают на использовании *Eclipse IDE*, потому что это один из самых больших и умных инструментов. Это очень сложная программа, а не просто блокнот для набора простого текста, поэтому она имеет такие полезные функции, как подсказывание или дополнение конструкций классов, методов, имеющихся в системе разработки и подключённых библиотеках, во время написания кода [1]. Однако эта система очень большая, при установке и настройке которой могут возникать трудности.

С другой стороны, есть более дружелюбный *Android Studio* как среда разработки мобильных приложений. В этой программе можно реализовать архитектуру *MVC* без каких-либо препятствий, с использованием *Java*, *XML* и графического представления макета, кнопок, и других элементов пользовательского интерфейса. *Android Studio* предназначена и создана для разработки программного обеспечения под операционную систему *Android* [4], в то время как в *Eclipse* можно создавать программы не только для мобильных устройств. Таким образом, многие программисты, от новичков до профессионалов, программируют в *Android Studio* из-за масштабируемости проектов, возможности эмулировать гаджеты или осуществлять отладку на реальных устройствах. Кроме того, он имеет интуитивно-понятный интерфейс, внутреннюю систему управления версиями, в отличие предыдущей среды разработки.

В *Visual Studio* также имеется возможность создания приложений под операционную систему *Android*, однако на установку *Xamarin* в *Visual Studio* требуются большие временные затраты, и могут возникнуть трудности, связанные с появлением ошибок при непосредственной установке и созданием проекта. Ещё одним немаловажным фактором отказа от *Xamarin* является сниженная производительность по сравнению с другими специализированными средами, поэтому дальнейшая работа с данным инструментом не продолжалась, а рассматривались предыдущие два варианта сред разработки.

Далее приведена сравнительная таблица возможностей *Android Studio* и *Eclipse* [5] (см. табл. 2). Наиболее подходящая среда разработки – *Android Studio*, созданная компанией Google специально для мобильной разработки под ОС *Android*. Среда позволяет проектировать графический интерфейс приложения с помощью встроенного графического дизайнера, постоянно обновляется и поддерживается, имеет множество положительных отзывов от программистов на различных форумах для разработчиков.

Особенности мобильной разработки

Почему именно мобильное приложение? Многие учёные отмечают, что с каждым годом число пользователей смартфонов динамично растёт, в основном с 2013 по 2018 год. Кроме того, многие эксперты считают ясным тот факт, что использование смартфона – это «время- и затратно-экономичное решение» [6]. Более того, количество доступных статистических, медицинских и спортивных приложений растёт, в магазинах мобильных приложений Google Play и App Store [7].

Таблица 2. Сравнительная таблица IDE Android Studio и Eclipse

Android Studio	Eclipse
<i>Общие характеристики</i>	
Интегрированные среды разработки	
Упрощённое представление структуры проекта, обеспечивая быстрый доступ к коду и файлам сборки	
Улучшенный Графический пользовательский интерфейс (GUI)	
Автодополнение кода	
Помогает писать чистый код (без ошибок)	
<i>Различия</i>	
Специализированная среда разработки Android-приложений	Широко используется для написания любых приложений на языке Java
Поддерживается Google	Google перестала осуществлять поддержку
Android поддерживается	Android поддерживается за счёт расширения Google ADT
Есть Gradle Build tool	Имеет ANT Build tool по умолчанию, но он устарел
Частые обновления для Android разработки	Нет частого обновления для Android разработки
Поддерживает только язык Java, а также язык разметки XML.	Поддерживает C, C++, C#, Java, JavaScript, Perl, PHP, Python и много других языков программирования
Разработана компанией Google	Разработана Eclipse Foundation
Специально разработана для мобильной Android-разработки	Разработана для создания различных видов приложений

Таким образом, выбор мобильной разработки оправдан. С другой стороны, процесс получения статистики огромного объёма данных может спровоцировать появление ещё одной проблемы, связанной с большими данными (Big data). В большинстве случаев «большие данные» классифицируются как «... набор данных, который является достаточно большим, чтобы выйти за рамки возможностей традици-

онных программных средств с точки зрения сбора, хранения и аналитики; большой объем данных, быстрый поток данных, различные типы данных и низкая плотность значения данных являются его специфическими характеристиками» (перевод) [8].

Поэтому разработчикам приходится тщательно продумывать алгоритмы анализа данных, чтобы избежать потери времени и ресурсов, например, памяти, производительности и качества получения результатов подсчёта. Однако есть хорошая сторона больших данных, которая состоит в охватывании множества возможных ситуаций, что повышает вероятность точного прогнозирования в различных ситуациях [9].

Одно из решений упрощения работы с большими данными предложили исследователи [10]. Они подчеркнули многофункциональный алгоритм анализа, который включает следующие этапы: экспресс-анализ, комплексный анализ, анализ «ЕСЛИ», и выдача рекомендаций. Этот алгоритм показывает хорошие результаты в разработанном приложении «Smart Mobile Health Navigator».

Следующим шагом в разработке интеллектуального программного обеспечения является использование карманных устройств для сбора данных в базу данных и их использования для подсчёта статистики. Эти устройства представляют собой компактный комплект приборов электроники, которые могут непрерывно собирать необходимые данные, получая детальную информацию о тренировке, здоровье человека. В целом, система карманных устройств являются набором, состоящим из трех частей: смартфон в качестве клиента, полученные по Bluetooth Post-данные (данные, передающиеся в теле запроса, для последующей обработки) и сервер (мобильные данные, доступ в интернет) [11].

В течение последних нескольких лет такие носимые устройства становятся все меньше, и этот факт даёт возможность использовать их даже в одежде [12]. Это чудо, что в настоящее время мы можем использовать такие устройства не только в виде коробки, пояса, часов, но и как эпидермальные устройства на основе химических датчиков. Может показаться фантастическим, но такая технология существует и обеспечивает получение данных из пота путём распознавания натрия, кальция и других элементов. Полученная информация приводит к объективным выводам о способности спортсмена к быстрому восстановлению во время тренировки [13].

Следовательно, на сегодняшний момент технологии мобильной разработки в спорте не стоят на месте, появляется широкий выбор

вспомогательных устройств для мониторинга тренировочного процесса, которые позволяют непрерывно получать очень большой объём информации, который, в свою очередь, может быть ценен в анализе спортивных достижений, а также направлен на улучшение результатов от начинающих спортсменов до профессионалов. Зная современные тенденции, будет полезным уметь обрабатывать различные показатели спортсмена и делать выводы по различным выявленным зависимостям.

Технологии и методы

Для того, чтобы обработать учётные данные спортсмена в мобильном приложении в виде графиков и агрегированных данных, необходимо реализовать технологии построения этих графиков и создать запросы к базе данных для получения агрегированных данных. Запросов требуют правильно написанную строку запроса и реализованный алгоритм работы с агрегированными данными. Процесс прорисовки графиков является одним из наиболее сложных.

Есть вариант написания модуля для приложения, который бы занимался построением этих графиков или использованием нейросетевых технологий. Этот вариант потребует больших временных и трудовых затрат, а также велика вероятность того, что система будет излишне перегружена для выполнения расчётов и прорисовки графиков. Чтобы избавиться от возможных ошибок можно выбрать готовую и протестированную библиотеку, подключить её к приложению и выполнять доступные построения, по необходимости дописывать к библиотеке что-то своё.

Существует множество доступных библиотек для построения графиков на Android, среди них такие как: `MapView` [14], `MPAndroidChart` [15], `WilliamChart` [15]. Перед тем как выбрать подходящую библиотеку было проведено их сравнение (см. табл. 3).

В итоге выбрана библиотека `MapView`. Она является достаточно простой в понимании, не требует большого количества времени на обучение работе с ней, её рекомендуют для небольших приложений. Кроме того, эта библиотека имеет официальный сайт с поддержкой, демонстрационными видео, обучающими примерами и возможностью использования основных видов диаграмм: линейная, точечная, столбчатая и их комбинации. Библиотеку можно найти в свободном доступе с полной документацией на `GitHub`.

**Таблица 3. Сравнительная таблица библиотек
для построения графиков**

Критерий	GraphView	MPAndroidChart	WilliamChart
Описание (документация)	достаточное, есть обучающие примеры	достаточное, есть обучающие примеры	недостаточное
Сайт	есть сайт, GitHub	Всё на GitHub	код на GitHub
Сложность	простая	сложная	простая
Возможности	Основные диаграммы, есть вспомогательная ось координат, возможность сохранения диаграмм или отправка друзьям в соц. сетях, добавление дат на ось абсцисс.	Очень много видов диаграмм, комбинированные диаграммы, вспомогательная ось, сохранение графиков в виде картинки, удобна для индивидуальной настройки.	Из описания: простые, многочисленные диаграммы, не осложняются полезными данными.
Рекомендации разработчиков	для небольших программ, для обучения	для продв. пользователей и сложных задач	мало отзывов
Степень разработанности	обновляется, популярна	обновляется, популярна	в разработке
Лицензия	+	+	+

Пример применения инструментов и технологий

Выбранные подходы, инструменты и технологии, описанные выше необходимо проверить в практической работе, чтобы понять, являются ли они подходящими для целого класса разработки мобильных приложений. Таким приложением будет «Дневник велосипедиста».

«Дневник велосипедиста» обычно ведётся в обычной тетради, где записываются данные о тренировке и самочувствии, что не очень удобно для оценки общего состояния спортсмена, отслеживания прогресса и дачи рекомендаций, т.к. для этого нужно делать много расчётов, которые занимают время и могут быть вычислены с ошибками. Поэтому есть необходимость создания специального приложения для этого, и лучше, чтобы это приложение было мобильным для удобства использования и данные всегда были рядом.

Данные для приложения должны храниться в локальной базе данных, которая будет создаваться для каждого экземпляра приложения, таким образом необходимость в подключении к интернету пропадает, что позволит работать с данными в любое время и в любом месте.

Далее необходимо создать класс, отвечающий за создание и работу с базой данных – это и будет представлять модель по паттерну MVC. После чего необходимо создать пользовательский интерфейс,

или представление приложения. Когда модель и представление готовы, необходимо приступить к созданию взаимодействия интерфейса с логикой. То есть нужно определить, что будет происходить при нажатии на ту или иную кнопку, какие данные и как нужно учитывать в базе данных. Также нужно выделить какие сообщения будет получать пользователь при возникновении исключительных ситуациях и при выполнении запросов к базе данных.

Выполнение описанных ранее действий привело к созданию мобильного приложения для ОС Android, которое обеспечивает учёт данных в системе (см. рис. 4) [17] и позволяет совершать некоторые аналитические операции, например, построение графика по пройденным расстояниям на тренировках (см. рис. 5), получать комбинированные графики за некоторый период (рис. 6), а также получать агрегированные данные (рис. 7) [18].

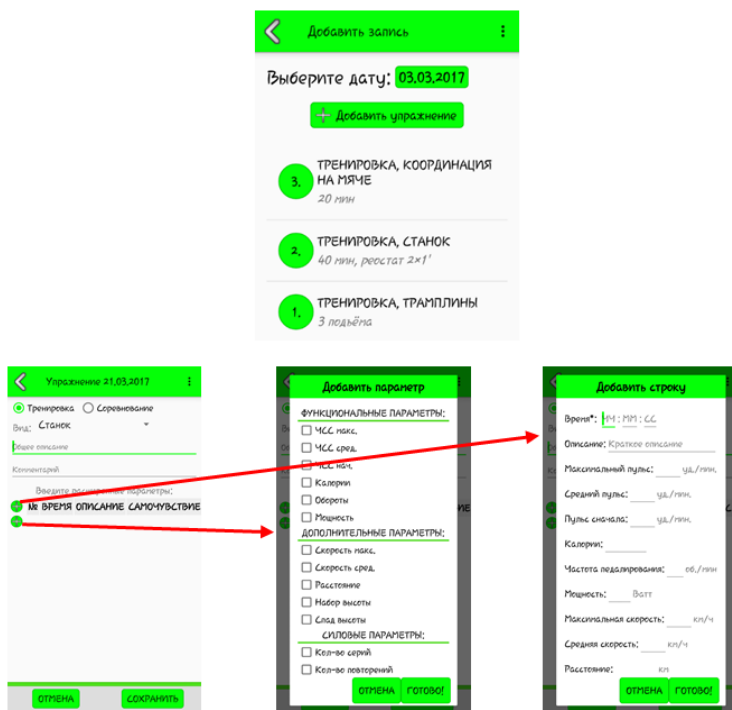


Рис. 4. Учёт данных в системе

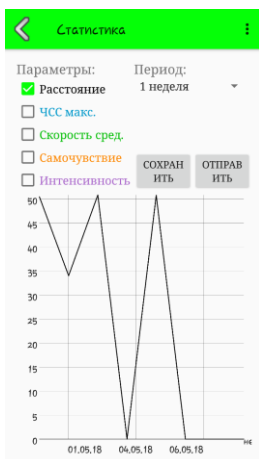


Рис. 5. График 1

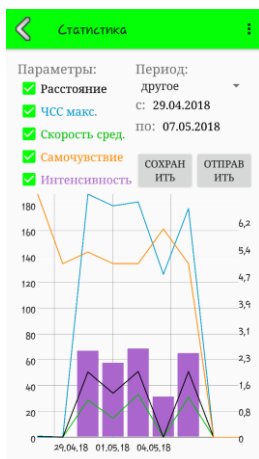


Рис. 6. График 2



Рис. 7. Агрегированные данные

Заключение

В статье рассмотрены паттерны проектирования архитектуры приложений (в т.ч. мобильных), произведено сравнение сред разработки, а также изучены технологии добавления аналитики в приложения по имеющимся данным.

В результате применения теоретических основ на практике, было получено полноценное мобильное учётное приложение для велосипедиста с возможностью получения графиков, построенных по занесённым в систему данным. При этом ни одна из компонент системы (модель, представление, контроллер) не мешает существованию других, выполняет только свои функции, что позволяет обеспечить бесперебойную работу пользователя с приложением.

Создание приложения осуществлялось по паттерну MVC в среде Android Studio с применением загруженной библиотеки GraphView. В дальнейшем приложение можно совершенствовать, просто добавляя необходимые функции.

Библиографический список

1. *Berentowicz, M., Zajac, P., Zabierowski W.* (2017). The use of mobile technology on the example of health and sports statistics application. *Polyana Svalyava (Zakarpattya): Ukraine MEMSTECH*, pp. 187-189.
2. Паттерны для новичков: MVC vs MVP vs MVVM. [Электронный ресурс]. Режим доступа: <https://habrahabr.ru/post/215605/>. Дата обращения: 19 февраля 2018 г.
3. *La, H.J., Kim, S.D.* (2010). Balanced MVC architecture for developing service-based mobile applications. *IEEE Computer Society*, pp. 292-299.
4. *Lamiaa, A. et al.* (2017). Real-time face detection and tracking on mobile phones for criminal detection. *IEEE*.
5. Difference Between Android Studio and Eclipse. (2018 January 26). Retrieved (date), from <https://www.differencebetween.com/difference-between-android-studio-and-vs-eclipse/>.
6. *García, B., Welford, J., Smith, B.* (2015). Using a smartphone app in qualitative research: the good, the bad and the ugly. *SAGE, Qualitative Research*, pp. 1-18, DOI: 10.1177/1468794115593335.
7. *Zhao, J., Freeman, B., Li, M.* (2016). Can mobile phone apps influence people's health behavior change? An evidence review. *Journal of medical internet research*, Vol. 18, iss. 11, e287, pp.1-12, DOI: 10.2196/jmir.5692.
8. *Zeng, J., Jia, J.* (2017). The impact of big data on school sports and competitive sports. *Chinese Automation Congress (CAC)*, pp. 596-599.
9. *Ahmadi, A., Mitchell, E., Richter, C., et al.* (2015). Towards automatic activity classification and movement assessment during a sports training session. *IEEE Internet of Things Journal*, Vol. 2, iss. 1, pp. 23-32, DOI: 10.1109/JIOT.2014.2377238.
10. *Dorosh, N., Kuchmiy, H., Boyko, O., et al.* (2016). Development the software applications for mobile medical systems based on OS Android. *TCSET*, pp. 808-810.
11. *Yanchapaxi, P., Tipantuña, C., Calderón, X.* (2017). Wearable system for monitoring of human physical activities. *IEEE*.
12. *Iervolino, R., Bonavolontà, F., Cavallari, A.* (2017). A wearable device for sport performance analysis and monitoring. *IEEE Conferences*, pp. 1-6.
13. *Seshadri, D., Drummond, C., Craker, J., Rowbottom, J., Voos, J.* (2017). New integrated technologies allow coaches, physicians, and

- trainers to better understand the physical demands of athletes in real time. *IEEE Pulse*, Vol. 8, iss. 1, pp. 38-43, DOI: 10.1109/MPUL.2016.2627240.
14. *Gehring, J.*, GraphView - open source graph plotting library for Android. (2018 January 26). Retrieved (date), from <http://www.android-graphview.org/>.
 15. *Jahoda, Ph.*, MPAndroidChart. (2018 January 26). Retrieved (date), from <https://github.com/PhilJay/MPAndroidChart>.
 16. *Bernardino, D.*, WilliamChart. (2018 January 26). Retrieved (date), from <https://github.com/diogobernardino/WilliamChart>.
 17. *Марквиреп В.Д.* Разработка приложения для учёта тренировочного процесса велосипедиста: курсовая работа. – Пермь: Пермский филиал Национального исследовательского университета «Высшая школа экономики». – 2017. – 85 с.: ил.
 18. *Марквиреп В.Д.* Разработка аналитического инструментария к мобильному учётному приложению для велосипедиста: выпускная квалификационная работа бакалавра. – Пермь: Пермский филиал Национального исследовательского университета «Высшая школа экономики». – 2018. – 69 с.: ил.

А.И. Миков ⁴

Кубанский государственный университет

alexander_mikov@mail.ru

ДИНАМИЧЕСКИЕ ГЕОМЕТРИЧЕСКИЕ ГРАФЫ В МНОГОСВЯЗНЫХ ОБЛАСТЯХ

Исследуются математические модели мобильных беспроводных компьютерных сетей в форме геометрических графов. Основное внимание уделено модели области расположения сети. Подробно исследовано влияние параметров двусвязной области на вероятностные характеристики связности случайных геометрических графов. Предложена и изучена модель многосвязной области и влияния ее коэффициента формы на возможности формирования маршрутов. Рассмотрен пример изменения динамического графа в многосвязной области. Основные положения иллюстрируются графиками, выполненными на базе статистического анализа.

Ключевые слова: случайный граф, связность, ad hoc сеть, маршрут, динамический граф, модель перемещения

Геометрические графы являются одной из математических моделей [1] компьютерных ad hoc сетей. Каждый такой граф порождается множеством точек области метрического пространства следующим образом: точке соответствует вершина графа; две вершины соединены ребром, если расстояние между ними (в метрике пространства) меньше некоторой заданной величины r . Обычно, в качестве метрического пространства берётся евклидова плоскость. Область может быть, либо не ограниченной, либо выбирается одна из простейших форм – квадрат, круг. Такая модель адекватно отображает гомогенные компьютерные сети, расположенные на открытых пространствах. Порождённый граф является обыкновенным неориентированным графом.

Однако, в реальных ситуациях ad hoc сети чаще располагаются на пересечённой местности, сочетающей и открытые пространства и ле-

⁴ Работа выполнена при финансовой поддержке РФФИ (грант №18-1-00359)

© Миков А.И., 2018

систую местность. Кроме того, приёмопередающие устройства беспроводной связи могут быть различных типов, с разными мощностями сигналов передатчиков и чувствительностью приёмников. При этом, как показано в [2], геометрический граф гетерогенной сети становится ориентированным.

В том случае, когда местность, в которой расположена сеть, содержит препятствия распространению сигнала (здания, горы), приходится в математической модели рассматривать уже многосвязные области, например, плоскость с «вырезанными» фигурами – кругами, квадратами, прямоугольниками. Соответственно, усложняется анализ характеристик порождаемого геометрического графа – его связности, количества мостов, отыскания чисел вершинной и рёберной связности графа.

В настоящей работе рассматриваются случайные геометрические графы в двусвязных и многосвязных областях, исследуются их вероятностные характеристики. В качестве двусвязной области используется модель «двух квадратов». В этой модели внешняя граница области представлена квадратом большего размера, а внутренняя граница (препятствие) – квадратом меньшего размера. Точки пересечения диагоналей квадратов совпадают, соответствующие стороны квадратов параллельны.

Целью исследования является отыскание вероятностных характеристик связности геометрических графов и их зависимости от параметров формы области расположения вершин графа.

Геометрические графы в двусвязных областях

Как показывает опыт исследования графов в односвязных выпуклых областях на плоскости, форма области (круг, квадрат, прямоугольник) не оказывает значительного влияния на результаты. Поэтому в качестве стандартной внешней границы области при вероятностном исследовании был взят квадрат со стороной, равной единице. Квадратное препятствие распространению сигналов передатчиков размещалось в центре квадрата, образуемом пересечением его диагоналей. Длина стороны препятствия b . Стороны препятствия параллельны сторонам внешнего квадрата. Таким образом, двусвязная область Q представляет собой квадратное «кольцо».

В этом «кольце» размещается множество $Dots = \{d_1, \dots, d_n\}$ из n точек – прообразов вершин графа (узлов компьютерной сети). С каждой точкой d_i связан открытый круг радиуса r с центром в этой точке. Этот круг определяет область, внутри которой отношение сигнал/шум

достаточно для того, чтобы принимать сигналы узла компьютерной сети, находящегося в точке d_i при отсутствии препятствий. Тройка $\{Q, Dots, r\}$ отображается в обыкновенный неориентированный граф $G = \{V, E\}$ следующим образом. Точка d_i отображается в вершину $v_i \in V$. Если между двумя точками d_i и d_j существует отрезок прямой, полностью принадлежащий области Q , и имеющий длину меньше r , то ребро $e_{i,j} \in E$. Полученный граф G называется (*статическим*) *геометрическим графом*.

Вместо множества $Dots$ рассмотрим множество $Traces = \{d_1(t), \dots, d_n(t)\}$. Здесь $d_i(t)$ – параметрические кривые, имеющие начальные точки $d_i(t_0)$. Кривой $d_i(t)$ соответствует вершина $v_i \in V$. Если для некоторого t между двумя точками $d_i(t)$ и $d_j(t)$ существует отрезок прямой, полностью принадлежащий области Q , и имеющий длину меньше r , то ребро $e_{i,j} \in E$ при данном значении параметра t . Построенная таким образом графовая структура называется *параметрическим геометрическим графом*. Если параметр t интерпретировать как физическое время, то параметрический граф можно назвать динамическим геометрическим графом.

При $b = 0$ площадь двусвязной области «двух квадратов» равна 1. Для лучшей сопоставимости результатов исследования размер площади сохраняется – при увеличении b и, соответственно, площади внутреннего квадрата, не входящего в область Q , сторона внешнего квадрата увеличивалась должным образом.

Анализ свойств отображения

Множество возможных геометрических конфигураций при конечном n имеет континуальную мощность. В то же время количество обыкновенных неориентированных графов с данным количеством вершин – конечное число, хотя, может быть, и очень большое. Таким образом, бесконечное количество конфигураций в качестве образа имеет один и тот же граф. Практическая невозможность описать подмножество конфигураций, отображающихся в данный граф, заставляет прибегать к вероятностно-статистическим методам исследования.

В работе рассмотрены случайные геометрические графы. Для статических геометрических графов берётся выборка из двумерного равномерного (на области Q) распределения вероятностей, которая и становится геометрической конфигурацией – множеством точек $Dots$. Координаты всех точек независимы. Для случайных динамических графов множество $Traces$ – многомерный случайный процесс. Далее,

описанным выше способом строится обыкновенный граф, свойства которого изучаются теоретико-графовыми методами, вычисляются характеристики, которые потом рассматриваются как реализации случайных величин. Множество реализаций случайного графа порождают множество реализаций его характеристик, обрабатываемых статистическими методами.

Ниже приведены результаты вероятностных исследований такой характеристики случайного графа, как количество рёбер. В частности, отыскивалось математическое ожидание количества рёбер для графов с $n = 5, 6, 10, 15, 20, 25, 30, 35, 40, 45$ с радиусами $r \in [0.1; 1.9]$ в областях Q с препятствиями с линейными размерами $b \in [0.1; 0.9]$. Семейство областей будем обозначать $Q(b)$.

На рис. 1 приведены зависимости математического ожидания количества рёбер в случайном статическом графе с 6 вершинами от радиуса r действия приёмопередатчиков для различных двусвязных областей $Q(b)$.

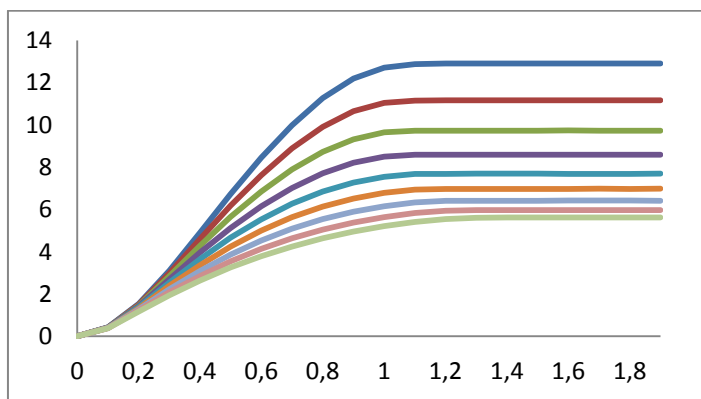


Рис. 1. Математическое ожидание количества рёбер, $n = 6$

Верхняя кривая соответствует области $Q(0.1)$. Следующая кривая – области $Q(0.2)$. Нижняя кривая построена для области $Q(0.9)$. В обыкновенном неориентированном графе с 6 вершинами может быть максимально 15 вершин. Предельное значение на графике 12.9. Таким образом, наличие 10-процентного (по линейному размеру) препятствия приводит в среднем к потере в графе двух рёбер. Четырнадцать и 15-рёберные реализации тоже существуют, но они маловероятны. При $b = 0.9$ в среднем теряется 9-10 рёбер. Хотя площадь области $Q(0.9)$ равна площади области $Q(0.1)$, но здесь уже играет роль ее форма –

ребра, в основном, вытягиваются в цепи или длинные циклы.

На рис. 2 приведены аналогичные кривые для случайных графов с 30 вершинами.

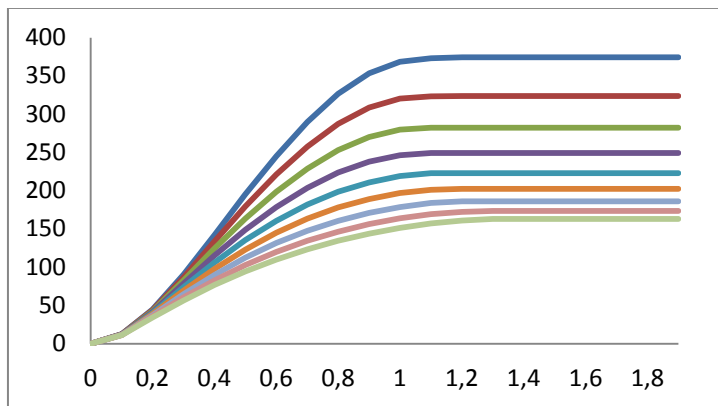


Рис. 2. Математическое ожидание количества рёбер, $n = 30$

На графиках рис. 2 хорошо видно, что при увеличении радиуса r математические ожидания Mq количества вершин в графе при всех значениях размеров b препятствия в центре области стремятся к предельным значениям. Эти предельные значения зависят от величины b , как показано на рис. 3.

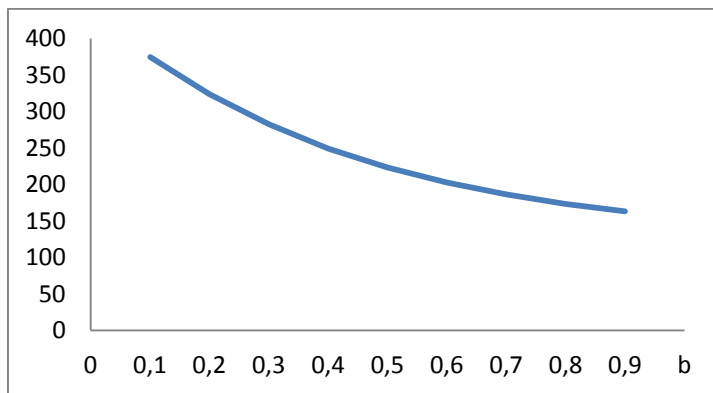


Рис. 3. Предельное математическое ожидание количества рёбер, $n = 30$

Зависимость $Mq = f(b)$ имеет гиперболический характер и может быть представлена эмпирической формулой $Mq = 200/(b + 0.462) + 16$. Эта формула хорошо согласуется с данными статистических экспериментов: в пределах $0.1 \leq b \leq 0.9$ относительная ошибка находится в пределах 1%, а при $b = 0$ относительная ошибка составляет 3%.

Количество рёбер в графе является важной характеристикой, но даже большое количество рёбер не гарантирует связности графа - важнейшей характеристики компьютерной сети. Поэтому отдельно изучалась вероятность того, что случайный граф с заданным количеством n вершин, расположенных в описанной двусвязной области, содержит маршруты, соединяющие любые пары вершин. Эта вероятность является функцией параметров r и b . На рис. 4 эта функция $P_c(r, b)$ приведена в виде семейства графиков, показывающих зависимость вероятности связности от величины радиуса r при фиксированных размерах препятствия b .

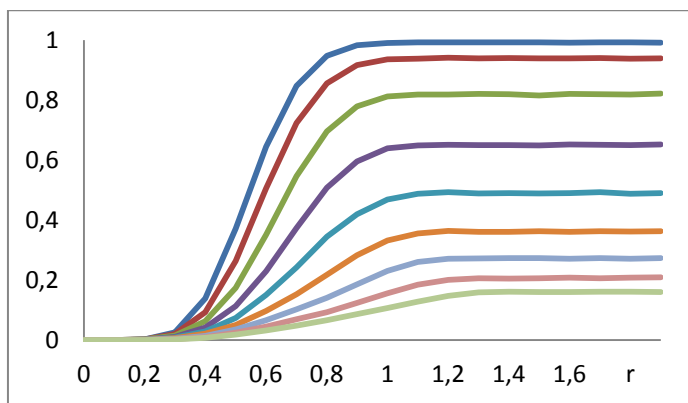


Рис. 4. Вероятность связности графа, $n = 5$

Как и на графиках рис. 1, 2 верхний график построен для $b = 0.1$, нижний график – для $b = 0.9$, остальные графики – для промежуточных значений b (0.2 ... 0.8). В целом зависимость является монотонно возрастающей по параметру r , и монотонно убывающей по параметру b . При увеличении радиуса характер кривых можно условно разделить на пять зон: практически нулевых значений, быстрого начального возрастания вероятностей, довольно большой практически линейный участок, замедляющийся переход к предельным значениям, участок не изменяющейся вероятности.

Имеются предельные значения вероятностей связности случайного графа $P(b) = \lim P_c(r, b)$ при $r \rightarrow \infty$. Зависимость $P(b)$ показана на рис. 5.

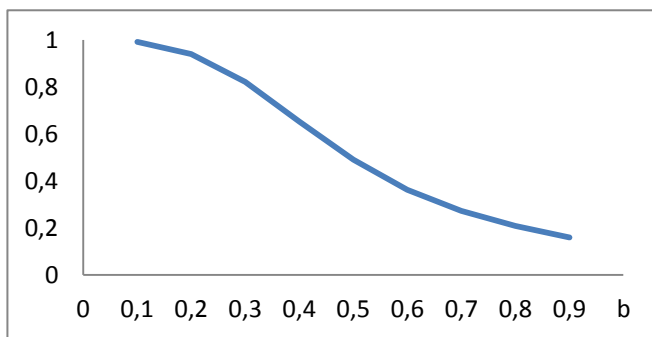


Рис. 5. Предельная вероятность связности, $n = 5$

График $P(b)$ имеет точку перегиба (вторая производная равна 0) при $b \approx 0.4$. Линейные участки имеют понижающиеся с ростом b коэффициенты наклона k . Значение $k(b)$ уменьшается от $k(0.1) = 2.52$ до $k(0.9) = 0.20$ с точкой перегиба $k(0.4) = 1.3$.

При большом количестве n вершин в геометрическом графе компьютерной сети линейный участок графика имеет очень большой коэффициент и вероятность связности увеличивается от нуля до предельного значения, определяемого величиной b , очень быстро, на коротком диапазоне изменения радиуса r . Например, при $b = 0.3$ вероятность связности не превышает 0.001 до $r = 0.2$ и имеет значение большее 0.999 при $r > 0.5$. В связи с этим на рис. 6 приведены вероятности $P_c(r, b)$ как семейство зависимостей от аргумента b при фиксированных значениях $r = 0.3$ (нижняя кривая); 0.4; 0.5.

Многосвязные области расположения геометрических графов вносят интересные изменения в их классификацию. В частности, известно, что отображение конфигураций точек плоскости, рассматриваемых как местоположения вершин геометрического графа, в множество обыкновенных неориентированных графов не является сюръективным. Существуют так называемые запрещённые графы (подграфы), которые не могут быть образами никаких геометрических конфигураций. Для остальных графов могут быть найдены [3] вероятности их появления в качестве образов.

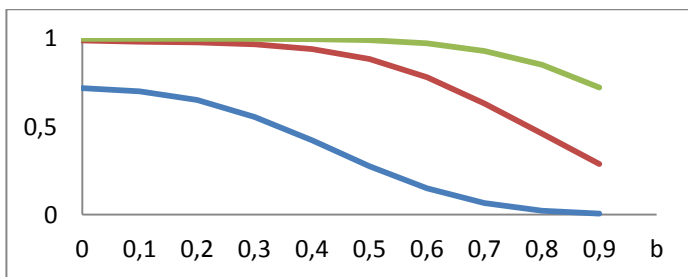


Рис. 6. Вероятность связности случайного графа, $n = 30$

Внесение препятствий в область расположения узлов компьютерной сети делает, по крайней мере, некоторые из запрещённых подграфов возможными. Для описанной выше двусвязной области «два квадрата» был проведён анализ возможности появления полного двудольного графа $K_{2,3}$ – запрещённого в односвязных областях. Оказалось, что такие графы возможны, хотя вероятность их как случайных графов невелика. В зависимости от величин r и b частоты появления графов $K_{2,3}$ при генерации 10^8 случайных графов оценивались в $10^{-6} - 10^{-5}$. Эти факты говорят о том, что свойства геометрических графов в многосвязных областях могут быть трудно предсказуемыми и значительно отличаться от свойств графов в односвязных областях.

Динамические графы в многосвязных областях

Для исследования динамических графов области с одним препятствием дают мало возможностей при анализе [4]. Одно препятствие оказывает локальное воздействие на граф, которое оказывает краткосрочное влияние на несколько вершин, перемещающихся в непосредственной близости от него. Более интересна картина с множественными препятствиями, образующими какую-либо регулярную структуру. На рис. 7 изображена такая структура, которую можно назвать «городскими кварталами».

Шесть вершин образуют при соответствующем радиусе связный граф. Если бы препятствий, создающих «радиотени» не было, то для связности графа потребовался бы существенно меньший радиус.

Обозначим, как и ранее, b – линейный размер препятствия. Кратчайшее расстояние между препятствиями обозначим a . Область и количество препятствий будем считать неограниченными как по горизонтали (ось x), так и по вертикали (ось y).

Тогда область расположения сети Q есть объединение полос:
 $-a/2 + i(a+b) \leq x \leq a/2 + i(a+b)$ при $i \in \mathbb{Z}$,
 $-a/2 + j(a+b) \leq y \leq a/2 + j(a+b)$ при $j \in \mathbb{Z}$.

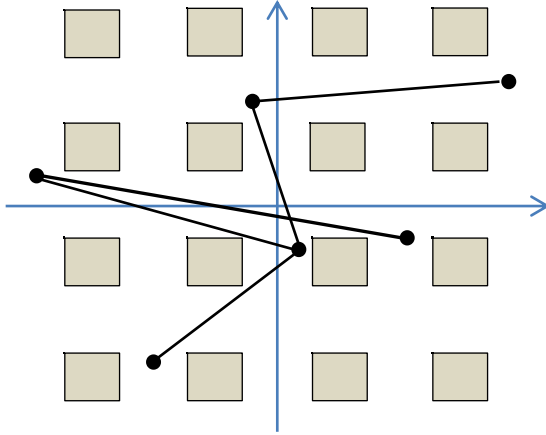


Рис. 7. Граф в многосвязной области, $n = 6$

Препятствие (i, j) задаётся одновременным выполнением четырёх неравенств:

$$(a/2 + i(a+b) \leq x \leq a/2 + b + i(a+b),$$

$$(a/2 + j(a+b) \leq y \leq a/2 + b + j(a+b),$$

Препятствие с номером $(0, 0)$ находится в начале I квадранта неограниченной многосвязной области Q .

Рассмотрим в этой области простой динамический граф, в котором n вершин перемещаются по серединам полос

$$-a/2 + i(a+b) \leq x \leq a/2 + i(a+b) \text{ при } i \in [-(n-1)/2, (n-1)/2],$$

с одинаковыми постоянными скоростями v . В начальный момент времени $t = 0$ вершины располагаются на оси x . Для симметрии возьмём нечетное n .

Поскольку в начальный момент времени расстояние между соседними вершинами равно $(a+b)$, то начальный граф $DG(t=0) = \neg K_n$ (вполне несвязный), если $r < (a+b)$, или P_n , если $(a+b) \leq r < 2(a+b)$, ..., K_n , если $r \geq (n-1)(a+b)$.

В момент времени $t_0 = (b/2)/v$ граф из-за попадания в зону препятствий станет вполне несвязным: $DG(t_0) = \neg K_n$, и будет таким оставаться до момента $t_1 = (b/2 + a)/v$, когда граф вновь вернётся к первоначальной конфигурации.

начальному состоянию. Таким образом, процесс изменения связности/ несвязности является периодическим.

Для случайных динамических графов, в которых вершины перемещаются по области произвольным образом, возможны различные варианты пересечения или не пересечения отрезка, соединяющего эти вершины с одним или несколькими препятствиями. От этого зависит наличие или отсутствие ребра и, в конечном счёте, связность динамического графа. На рис. 8 приведены графики зависимостей вероятности пересечения от длины r отрезка.

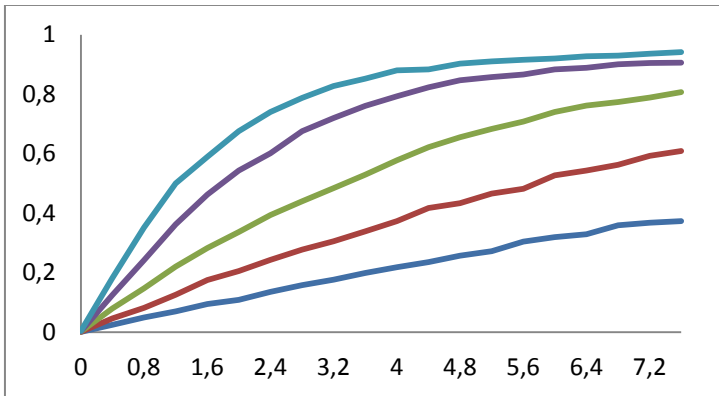


Рис. 8. Вероятность пересечения отрезка с препятствием

Для всех графиков величина «коридора» $a = 1$. Величины b препятствий меняются: нижнему графику соответствует значение $b = 0.05$. В этом случае зависимость $P(r)$ близка к линейной: $P(r) = \varphi \times r$, где $\varphi = b/a$ – коэффициент формы. Следующие графики приведены для значений $\varphi = 0.1$; $\varphi = 0.2$; $\varphi = 0.5$. Верхний график соответствует $\varphi = 1.0$.

Расчёты показывают, что вероятности существования связей между узлами сети в значительной мере определяются коэффициентом формы области расположения сети.

Многосвязные области являются естественной моделью местности, в которой приходится функционировать компьютерным ad hoc сетям. Поэтому их изучение, выявление особенностей отображения геометрических конфигураций в статические и динамические графы является интересной математической и важной практической задачей.

Библиографический список

1. *Penrose M.* Random geometric graphs: Oxford Studies in Probability / *M. Penrose*. – Oxford: Oxford University Press, 2003.
2. *Миков А.И.* Динамические геометрические графы мобильных ad hoc сетей / *А.И. Миков* // Информатизация и связь. – 2017. – № 2. – С. 66–70.
3. *Миков А.И.* Стохастические характеристики разрезов в графах ad hoc сетей / *А.И. Миков* // Информатизация и связь. – 2014. – № 1. – С. 76–79.
4. *Díaz J.* On the relation between graph distance and Euclidean distance in random geometric graphs / *J. Díaz, D. Mitsche, G. Perarnau, X. Pérez-Giménez* // Advances in Applied Probability. – 2016. – № 3. Pp. 848–864.

М.А. Плаксин

Национальный исследовательский университет
«Высшая школа экономики»
(Пермский филиал)

Пермский государственный национальный
исследовательский университет

mapl@list.ru

ОБ ОДНОМ МЕТОДЕ ОЦЕНКИ КАЧЕСТВА УПРАВЛЕНИЯ ОРГАНИЗАЦИОННОЙ СИСТЕМОЙ

В статье описана методика применения метода диверсионного анализа для оценки качества управления организационной системой. Для выявления потенциальных недостатков и оценки их значимости используется коллективная анонимная экспертиза и статистическая обработка экспертных оценок. В результате диверсионный анализ оказался полезен для оценки качества управленческого процесса и выявления путей его совершенствования. В качестве примера рассмотрено применение описанной методики к оценке качества учебного процесса на факультете бизнес-информатики НИУ ВШЭ-Пермь. Но не видно препятствий к применению предложенной методики в других случаях.

Ключевые слова: качество управления, метод диверсионного анализа, коллективная анонимная экспертиза, статистический анализ

Введение

Данная статья посвящена возможности применения метода диверсионного анализа для оценка качества управления организационной системой. В качестве примера рассматривается применение метода для оценки качества образовательного процесса в Пермском филиале Высшей школы экономики. Но объектом исследования в данном случае является не образовательный процесс, а пригодность диверсионного анализа для управления организационной системой.

Метод диверсионного анализа (ДА) был разработан в 70-е годы XX в. Автор метода – будущий Мастер ТРИЗ Б.Л. Злотин [1, 2, 5, 6, 7]. Исторически метод создавался как инструмент, применяемый на аналитической стадии функционально-стоимостного анализа [3]. В Западной Европе и США он известен под названием «Упреждающее определение отказов» – Anticipatory Failure Determination (AFD).

Начальное назначение ДА – выявление причин брака в производственном процессе. Причём поиск причин брака можно вести двумя способами. Назовём их условно «прицельная стрельба» и «стрельба по площадям».

«Прицельная стрельба» выглядит следующим образом. Пусть есть некий производственный процесс П. При правильном выполнении на выходе оно должен выдавать продукт А. Но в реальности он выдаёт продукт В, отличный от А. То есть производит брак. То есть содержит какие-то погрешности. Стандартный подход к исправлению процесса П заключается в решении исследовательской задачи: надо исследовать процесс П и найти причины, по которым он производит брак. Диверсионный анализ предлагает заменить исследовательскую задачу на задачу изобретательскую. Он предлагает не искать причины появления брака, а специально сконструировать производственный процесс таким образом, чтобы его результатом стал продукт В. Когда эта задача будет решена – требуемые условия производства бракованной продукции найдены – нужно проверить, до какой степени эти условия реализованы в реальном технологическом процессе. Если они реализованы в результате нарушения технологии, эти нарушения надо устранять. Если эти условия являются частью технологии, надо менять технологию.

Второй способ применения диверсионного анализа – это «стрельба по площадям». В этом случае ДА применяется не для того, чтобы определить причину появления отказов конкретного вида, а для того, чтобы отыскать слабые места и потенциальные недостатки технологического процесса вообще. В этом случае диверсионный анализ проходит в два этапа. На первом этапе проводится целенаправленный поиск возможных нарушений идеального технологического процесса, которые могут понизить качество выпускаемой продукции. Причём эти нарушения не должны быть явными, должны как можно лучше вписываться в правильную технологическую цепочку. На втором этапе проводится анализ реального технологического процесса на предмет того, до какой степени найденные «диверсии» – способы понизить качество продукции – уже «внедрены в производство», уже реализуются на практике. Как показывает опыт, на реальных производствах число «внедрённых» «диверсий» достигает 80% теоретически возможных.

Это второй вариант использования ДА – «стрельба по площадям» – и породил англоязычное название метода – Anticipatory Failure Determination: предсказание сбоев в технологическом процессе до того, как они произошли в реальности.

Изначально диверсионный анализ предназначался для работы с техническими системами, для совершенствования технологических процессов. Нас будет интересовать возможность использования этого метода для усовершенствования процесса управления организационной системой. В качестве примера для исследования была выбрана организация учебного процесса на кафедре информационных технологий Пермского филиала НИУ Высшая школа экономики (НИУ ВШЭ-Пермь).

Отметим два важных отличия управленческого процесса от технологического.

1. Точность измерения результата. Результат технологического процесса, как правило, определён достаточно строго. Выявить отклонения можно с помощью измерения хорошо определённых параметров (размера, веса, цвета, прочности, скорости и т.п.). Для измерения значений этих параметров существуют специальные инструменты и общепринятые объективные методики. Для управленческого процесса такие чётко определённые параметры очень часто отсутствуют. Набор параметров, по которым оценивается качество результата, формируется экспертным путём. Естественно, для выбранного экспертным путём набора параметров, отсутствуют объективные измерители. Оценивать их значение приходится также экспертным путём.

2. Скорость обратной связи. Зачастую скорость обратной связи в технологическом процессе много выше, чем в организационном. Так в рассматриваемом примере – в организации учебного процесса – для многих параметров цикл обратной связи равен учебному году.

Опыт применения диверсионного анализа к оценке учебного процесса в НИУ ВШЭ-Пермь

Начиная с 2012/13 уч.г. в течение трёх лет на факультета бизнес-информатики Пермского филиала НИУ Высшая школа экономики проводился эксперимент по применению метода диверсионного анализа для поиска путей совершенствования образовательного процесса [4]. Непосредственно в то время эксперимент преследовал две цели:

1. Проверить, насколько возможно применение метода для анализа учебного процесса в вузе.

2. Собрать информацию для совершенствования учебного процесса на одном из факультетов ВШЭ.

В данном случае диверсионный анализ применялся во втором варианте – в виде «стрельбы по площадям» – для поиска недостатков в учебном процессе факультета вообще.

Для проведения обоих этапов исследования было использовано анкетирование множества экспертов и статистическая обработка собранных анкет.

К выбору экспертов были предъявлены три требования:

1. Привлечение экспертов должно было стоить достаточно дешево.

2. Эксперты должны были быть заинтересованы в качестве учебного процесса.

3. Эксперты должны были иметь свежий взгляд на оценку качества учебного процесса.

Для оценки качества учебного процесса было решено взглянуть на него не со стороны преподавателя, а со стороны учащегося. Для сбора «диверсионных идей» был задействован традиционный контингент большинства университетских проектов – студенты. Выглядело это следующим образом.

Студенты разных курсов были ознакомлены с понятием «диверсионный анализ» (в большинстве случаев, в форме краткой лекции). После чего им было предложено написать два небольших эссе на темы «Каким образом организовать учебный процесс так, чтобы выпускники факультета как можно меньше удовлетворяли потребностям работодателя?» и «Каким образом организовать набор на факультет так, чтобы набрать на факультет самых неподходящих абитуриентов?».

Ожидалось, что ответы на первый вопрос помогут выявить слабые места в организации учебного процесса, а ответы на второй дадут информацию о профориентационной работе в школе. Первые ожидания оправдались, вторые – нет. Почти все идеи второго эссе касались процедуры подачи документов, правил зачисления и других аспектов, регулируемых на федеральном уровне. Для совершенствования работы факультета эти идеи были совершенно бесполезны. Поэтому дальнейший их анализ не проводился. Второй этап был проведён только для «диверсий», предложенных в первом эссе.

Были составлены перечни «диверсионных идей», предложенных студентами. При наличии нескольких формулировок одной идеи в перечень заносилась одна из них. Для каждой идеи был подсчитан уровень ее популярности – количество экспертов, ее предложивших.

Характеристики первого этапа по годам приведены в табл. 1.

Таблица 1. Характеристики первого этапа диверсионного анализа факультета бизнес-информатики НИУ ВШЭ-Пермь по годам

Характеристика	Год		
	2012/13	2013/14	2014/15
Количество участников	68	61	104
Общее количество «диверсий»	314	497	639
Количество оригинальных идей	117	158	70
Количество повторов «верхних» 30 идей	223	307	536
Количество повторов «верхних» 11 идей	155	200	300

Количество предложенных на первом этапе «диверсий» могло превышать полторы сотни. Дальнейшая работа с таким количеством идей была физически невозможна. Но в этом и не было необходимости. Во-первых, часть идей выходила за рамки компетенции факультета (например, касалась правил назначения стипендии). Во-вторых, значимость идей была весьма различна. В качестве измерителя значимости «диверсии» было решено использовать ее «популярность»: количество человек, предложивших эту идею. Для дальнейшего анализа отбирались пять-семь наиболее частых предложений. Реально эти пять-семь «диверсий» явились обобщением и переработкой значительно более обширного набора (двух-трех десятков) студенческих идей, имеющих разную формулировку, но близких по сути.

В табл.2 приведены наиболее популярные «диверсионные идеи» 2014/2015 уч.г.

При повторении процедуры ДА в течение нескольких лет оказалось, что список самых значимых (самых популярных) «диверсий» достаточно стабилен. В табл.3 перечислены наиболее популярные идеи за три года (2012/13 – 2014/15 уч.гг.).

На втором этапе экспертам было предложено оценить, до какой степени выбранные «диверсии» уже реализованы в учебном процессе факультета.

Выглядело это так.

Студентам были предложены анкеты (свои для каждого курса), построенные по матричному принципу. Столбец матрицы соответствовал одной из выбранных «диверсионных идей», строка – одной из изученных студентами дисциплин.

**Таблица 2. Наиболее популярные «диверсионные идеи»
2014/2015 уч.г.**

Идея	Количество голосов	Доля, %
1. Больше лекций, меньше практики	89	14
2. Не проверять регулярно знания студентов	46	7
3. Низкие требования (ставить зачёт «за красивые глаза», за посещение, мягко относиться к срокам сдачи и выполнению требований).	45	7
4. Преподаватель не имеет опыта по своему предмету, читает лекции монотонно и скучно.	41	6
5. Разрешить студентам не ходить на пары, ввести свободное посещение.	36	3
6. Нет командных работ (в группах).	25	4
7. Преподаватель не должен консультировать студентов	23	4
8. Увеличить количество часов на непрофильные предметы	23	4
9. Перестать выплачивать стипендию	22	3
10. Перестать обновлять оборудование и программное обеспечение	22	3

**Таблица 3. Наиболее популярные «диверсионные идеи»
за три года**

«Диверсионная идея»	2012/13	2013/14	2014/15
1. Много теории, но мало практики	+	+	+
2. Не проверять регулярно знания студентов (нет домашних заданий, микроконтролей и т.п.)	+	+	+
3. Разрешить студентам не ходить на пары, ввести свободное посещение	+	+	+
4. Низкие требования (низкие требования к оцениванию, низкая сложность заданий, отсутствие дедлайнов, много пересдач)	+		+
5. Преподаватель не имеет опыта по своему предмету, читает лекции монотонно и скучно	+	+	+
6. Преподаватель не должен консультировать студентов		+	+
7. Нет командных работ (в группах)		+	+
8. Использовать устаревшее программное обеспечение и методики преподавания		+	

От студентов требовалось:

1) оценить важность каждой «диверсии» (каждого из указанных недостатков организации учебного процесса);

2) оценить степень проявления данного недостатка в процессе преподавания данной дисциплины.

В обоих случаях использовалась шкала от 0 до 10: 0 – минимальная важность недостатка, 10 – максимальная; 0 – отсутствие данного недостатка при преподавании данной дисциплины, 10 – максимальное его проявление.

Анкеты, заполненные студентами одного курса, консолидировались: по каждой позиции вычислялась среднее арифметическое. После этого вычислялись следующие показатели (см. табл. 4):

1) максимально возможный на данном потоке негативный рейтинг дисциплины;

2) негативный рейтинг, полученный каждой из дисциплин.

Таблица 4. Схема таблицы для оценки степени проявления недостатков в процессе преподавания различных дисциплин

Дисциплина	Недостаток					Негативный рейтинг абсолютный	Негативный рейтинг относительный
	1	2	3	4	5		
	v_1	v_2	v_3	v_4	v_5	$\sum_i v_i \cdot 10$	
1	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	$\sum_i v_i \cdot p_{1i}$	$\frac{\sum_i v_i \cdot p_{1i}}{\sum_i v_i \cdot 10}$
2	p_{21}	p_{22}	p_{23}	p_{24}	p_{25}	$\sum_i v_i \cdot p_{2i}$	$\frac{\sum_i v_i \cdot p_{2i}}{\sum_i v_i \cdot 10}$
3	p_{31}	p_{32}	p_{33}	p_{34}	p_{35}	$\sum_i v_i \cdot p_{3i}$	$\frac{\sum_i v_i \cdot p_{3i}}{\sum_i v_i \cdot 10}$

Здесь v_i – важность i -го недостатка, p_{ij} – степень проявления i -го недостатка в процессе преподавания j -й дисциплины

Максимально возможный негативный рейтинг вычислялся как сумма произведений коэффициента важности этого недостатка на максимально возможную степень проявления этого недостатка (т.е. на 10). Поскольку важность недостатков по-разному оценивалась студентами

разных курсов, максимально возможный негативный рейтинг для каждого курса был своим.

Негативный рейтинг каждой дисциплины вычислялся как сумма произведений коэффициента важности недостатка на степень проявления этого недостатка в процессе преподавания этой дисциплины. Поскольку важность недостатков студентами каждого курса была оценена по-своему, абсолютная величина негативного рейтинга оказалась недостаточно информативна как для целей оценки, так и для целей сравнения. Поэтому абсолютные величины были пересчитаны в относительные. Для каждой дисциплины был вычислен процент полученного ею негативного рейтинга от максимально возможного негативного рейтинга.

В подвал таблицы были добавлены средние, максимальные и минимальные значения по всем графам.

Пример обсуждаемой таблицы представлен в табл.5.

Таблица 5. Фрагмент таблицы для оценки степени проявления недостатков в процессе преподавания различных дисциплин

Дисциплина	Недостаток					Негативный рейтинг абсолютный	Негативный рейтинг относительный
	Мало практики	Вести свободное посещение	Не проверять знания	Низкие требования	Преподаватель не имеет опыта		
	9,17	3,67	8,50	9,33	9,00	396,67	100%
Безопасность жизнедеятельности	6,75	1,50	1,63	1,75	1,00	106,52	26,85%
Программирование	3,13	1,13	1,75	0,63	0,38	56,85	14,33%
Компьютерная графика	0,63	1,88	0,63	1,13	0,00	28,42	7,16%
Физическая культура	0,00	3,86	0,86	0,00	0,00	21,43	5,40%
Среднее	2,29	1,98	1,15	0,69	0,63	50,19	12,65%
Max	6,75	3,86	2,25	1,75	2,75	106,52	26,85%
Min	0,00	0,57	0,38	0,00	0,00	21,43	5,40%

Оценка качества управленческого процесса с помощью диверсионного анализа

Рассмотрим, как собранную в ходе ДА информацию можно использовать для оценки качества управленческого процесса.

Интерес представляют следующие факторы:

- 1) индивидуальные оценки,
- 2) групповые оценки разного уровня (факультет, курс, дисциплина),
- 3) анализ изменения оценок от года к году,
- 4) сопоставление оценок друг с другом.

Полезность индивидуальных оценок для конкретного преподавателя очевидна. Это конкретный сигнал о неполадках или информация об имеющемся «запасе прочности». Например, достаточно большое значение для диверсии «Низкие требования» покажет, что дисциплина не воспринимается студентами как слишком сложная.

Рассмотрим примеры трёх других факторов.

В табл.6 приведена средний негативный рейтинг факультета за три года. Все это время он колеблется в районе 22-25%. Видимо, эта величина достаточно точно отражает состояние факультета. Положение обстоит неплохо. Но и возможности для улучшения отнюдь не исчерпаны!

Таблица 6. Средний негативный рейтинг факультета

Год	2012/13	2013/14	2014/15
Средний рейтинг, %	23,91	22,02	24,68

В табл.7 и на рис.1 негативный рейтинг курсов сопоставляется с рейтингом факультета. (Сокращения «I БИ» и «I ПИ» обозначают I курс направлений бизнес-информатика и программная инженерия.) Легко заметить, что положение дел на курсе I ПИ вдвое лучше, чем в среднем по факультету, а на III курсе наоборот, существенно хуже. Мы имеем индикатор, сигнализирующий о необходимости управленческой реакции.

Таблица 7. Рейтинг курсов в сопоставлении с рейтингом факультета

Курс	Факультет	I БИ	I ПИ	II	III	IV
Средний рейтинг, %	23,91	18,19	12,65	24,99	37,24	26,46

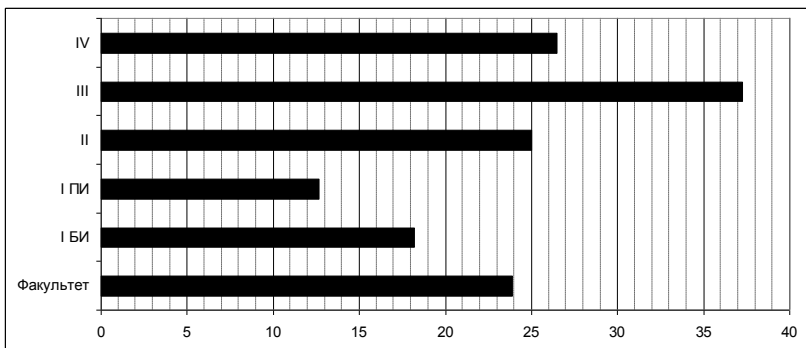


Рис. 1. Рейтинг курсов в сопоставлении с рейтингом факультета

В табл.8 и на рис.2 приведены оценки качества одной и той же дисциплины, данные студентами разных курсов. (На диаграмму добавлен ещё и средний рейтинг факультета.) Очевидно, что представление третьекурсников резко отличается от представления всех остальных курсов. Разница со вторым курсом составляет более 14 раз! Судя по этому индикатору, требуется срочное управленческое вмешательство! А вот на II курсе ситуация много лучше среднефакультетской!

Таблица 8. Рейтинг одной и той же дисциплины на разных курсах

Курс	I	II	III	IV
Рейтинг, %	10,5	4,5	65	17

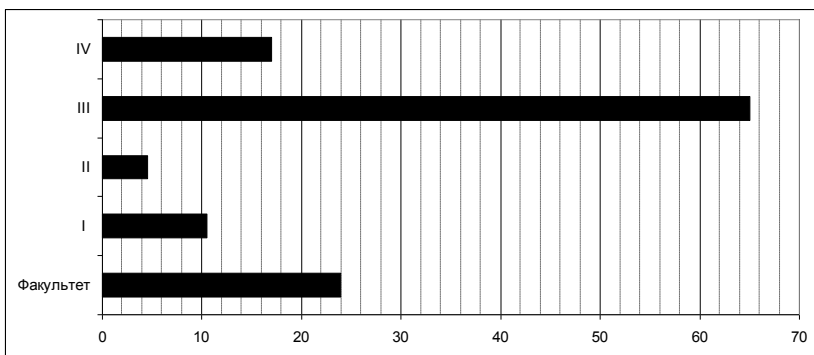


Рис. 2. Рейтинг одной и той же дисциплины на разных курсах

В табл.9 и на рис.3 отслеживается изменение оценок одной дисциплины в течение трёх лет. В целом положение обстоит неплохо, нега-

тивные рейтинги достаточно низкие. Но несколько моментов настораживают. Оценка диверсии «Больше лекций, меньше практики» упорно ползёт вверх и за три года выросла вдвое. Ещё хуже дело с диверсией «Низкий уровень требований». За три года мы имеем пятнадцатикратный рост! В восемь раз выросла оценка диверсии «Набирать неопытных преподавателей». Либо в 2014/15 уч.г. курс вёл другой преподаватель. Либо «опытность» преподавателя в восприятии студентов упала в восемь с лишним раз! А вот консультации стали более регулярными!

Таблица 9. Изменение оценок одной дисциплины за несколько лет

Недостаток	2012-13	2013-14	2014-15
Больше лекций, меньше практики	1,38	2,64	2,90
Свободное посещение занятий	1,75	1,36	1,50
Не проверять регулярно знания студентов	1,38	2,00	1,13
Низкий уровень требований	0,25		3,80
Набирать неопытных преподавателей	0,25	0,45	2,07
Не проводить консультаций		4,00	2,67
Нет работы в команде		1,73	1,47
Использовать устаревшее ПО и методики		0,27	

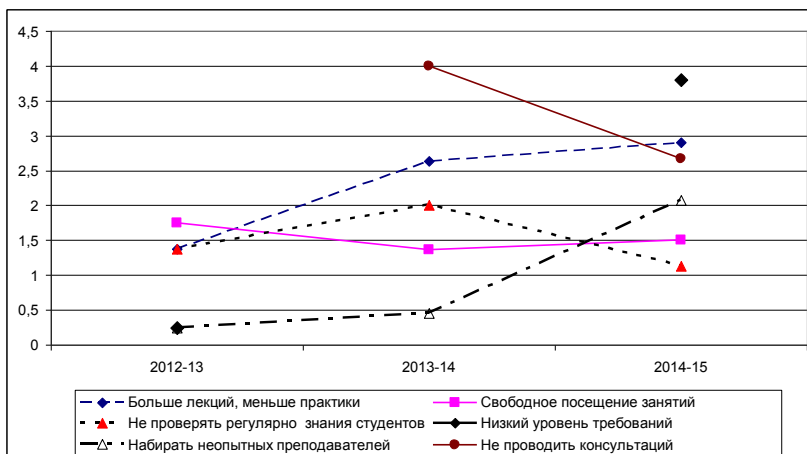


Рис. 3. Изменение оценок одной дисциплины за несколько лет

В табл.10 показано отношение к одним и тем же недостаткам со стороны разных курсов. Для простоты сравнения в таблицу добавлены строка и графа с суммой оценок по графам и строкам.

Обращает на себя внимание разница сумм первой и второй граф – оценок, данных курсами I БИ и I ПИ. Первокурсники - программные инженеры в три раза более чувствительны к недостаткам, чем их коллеги с бизнес-информатики! Интересно, в чем причина такой обострённой чувствительности? Или наоборот, в чем причина низкой чувствительности первокурсников – бизнес-информатиков? Ведь остальные курсы явно склоняются к мнению программных инженеров.

Наименьшее неприятие вызывает диверсия «Свободное посещение занятий». Остальные четыре недостатка идут примерно вровень. Но *primus inter pares* все-таки «Много теории, мало практики».

Таким образом, мы и здесь имеем целый ряд индикаторов, важных для принятия управленческих решений.

Таблица 10. Отношение к одним и тем же недостаткам со стороны разных курсов

Недостаток	Курс					Сумма
	I БИ	I ПИ	II	III	IV	
Много теории, но мало практики	5,23	9,17	8,33	5,83	8,40	36,96
Свободное посещение занятий	2,31	3,67	4,30	5,33	4,90	20,51
Не проверять регулярно знания студентов	2,08	8,50	6,60	6,50	6,50	30,18
Низкие требования	1,62	9,33	8,40	5,33	7,80	32,48
Набирать неопытных преподавателей	2,69	9,00	8,10	5,00	9,40	34,19
Сумма	13,93	39,67	35,73	27,99	37,00	

Заключение

Диверсионный анализ изначально создавался как метод совершенствования технологических процессов. Мы рассмотрели методику, которая позволяет применить его для оценки качества управления организационной системой. Поскольку результаты работы организационной системы определены гораздо менее точно, чем системы технической, предложенная методика опирается на экспертную оценку. Для повышения объективности, применяется коллективная анонимная экспертиза и статистическая обработка экспертных оценок. В результате диверсионный анализ оказался полезен для оценки качества управленческого процесса и выявления путей его совершенствования.

В качестве примера рассмотрено применение описанной методики к оценке качества учебного процесса на факультете бизнес-информатики НИУ ВШЭ-Пермь. Но при этом не использовалось никаких особенностей именно данной системы. Так что не видно препятствий к применению предложенной методики в других случаях.

Библиографический список

1. *Злотин Б.Л., Зусман А.В.* Решение исследовательских задач. Кишинев: МНТЦ «Прогресс, Картя Молдовеняскэ», 1991.
2. *Злотин Б.Л., Зусман А.В.* Методика прогнозирования чрезвычайных ситуаций, вредных и нежелательных явлений. – Кишинев: МНТЦ "Прогресс", 1991. [Электронный ресурс] [Режим доступа: <http://metodolog.ru/00891/00891.html> Проверено 07.11.2018]
3. *Литвин С.С., Герасимов В.М.* Основные положения методики проведения функционально-стоимостного анализа. Методические рекомендации. – М.: Информ-ФСА, 1991.
4. *Плаксин М.А.* Применение диверсионного анализа для совершенствования образовательного процесса. // Пятая Международная конференция «Системный анализ и информационные технологии» САИТ-2013 (19–25 сентября 2013 г., г. Красноярск, Россия): Труды конференции. В 2-х т. – Т.2. – Красноярск: ИВМ СО РАН, 2013. С.171-179.
5. AFD (Anticipatory Failure Determination): [Электронный документ] (<http://www.ideationtriz.com/>). Проверено 07.11.2018.
6. *Kaplan S., Visnepolschi S., Zlotin B., Zusman A.* New tools for failure & risk analysis. Anticipatory Failure Determination (AFD) and The Theory of Scenario Structuring / Ideation International Inc., 1999, 2005. – USA: ISBN 1-928747-0-51. [Электронный документ] [Режим доступа: <http://whereinnovationbegins.net/build/publication.asp>] (Проверено 07.11.2018).
7. *Visnepolschi S.* How to Deal with Failures (The Smart Way) / Ideation International Inc., 1999, 2005. – USA: ISBN 1-928747-0-51. [Электронный документ] [Режим доступа: <http://whereinnovationbegins.net/build/publication.asp>] (Проверено 07.11.2018).

М.С. Сахипова^{1,2}, Л.Н. Лядова^{1,2}

¹Пермский государственный национальный
исследовательский университет
MSSakhipova@hse.ru

²Национальный исследовательский университет
«Высшая школа экономики»
(Пермский филиал)
LNLyadova@gmail.com

О ПОДХОДЕ К РЕШЕНИЮ ЗАДАЧИ АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ АРХИТЕКТУРЫ ПРЕДПРИЯТИЯ

В статье анализируются инструменты, используемые при проектировании архитектуры предприятия, описываются их характеристики. Обосновывается вывод о том, что основной проблемой, затрудняющей процесс проектирования, приводящей к повышению трудоёмкости работы системных аналитиков и инженеров, является отсутствие средств интеграции используемых инструментов. Предлагается подход к решению задачи создания системы автоматизации проектирования архитектуры предприятия и описывается общая архитектура разрабатываемых программных средств.

Ключевые слова: архитектура предприятия, проектирование, система автоматизации проектирования, архитектура системы.

Введение

Автоматизация бизнес-процессов, а также их моделирование и анализ являются залогом успешного существования и функционирования компании. Реализация данных процессов требует как квалифицированного подхода, так и понимания всех нюансов деятельности предприятия в конкретной предметной области. Впоследствии специфика области деятельности воплощаются в описаниях бизнес-процессов, а также моделях предприятия (модели бизнес-архитектуры, технологической инфраструктуры, архитектуры приложений).

Все современные организации должны быстро реагировать на изменения внешней среды, рынка и технологий. Гибкость и адаптивность бизнеса реализуются через оперативное изменение организационной

структуры, системы целей и показателей, реинжиниринг бизнес-процессов, совершенствование технологической инфраструктуры [3]. Изменения в компании требуют согласованной работы различных подразделений организации и для получения положительного эффекта необходимо представлять компанию как единую систему. Обеспечить целостное понимание устройства компании для различных категорий специалистов могут помочь визуальные модели архитектуры предприятия.

Для повышения эффективности использования ресурсов предприятия необходимо связать бизнес-цели организации с архитектурой корпоративной информационной системы (ИС), а для достижения этой цели, нужна грамотно построенная архитектура предприятия (АП), учитывающая как слой бизнес-архитектуры, так и системную архитектуру предприятия. Она поможет сфокусировать информационные технологии (ИТ), внедрённые в компании, на целях и задачах конкретного бизнеса. В организации с правильно настроенной архитектурой предприятия гораздо проще внедрять новые автоматизированные функции и управлять изменениями с минимальными затратами времени и ресурсов [4].

Таким образом, актуальной становится задача создания средств автоматизации проектирования архитектуры предприятия, интегрирующей различные инструменты моделирования и анализа, применяемые для решения задач как на уровне системной, так и бизнес-архитектуры.

Основные понятия

Архитектура предприятия – инструмент организации, который позволяет планировать изменения в бизнес-процессах и структуре предприятия, особенно в части, отвечающей за использование ИТ. В архитектуру предприятия включаются представления о бизнес-архитектуре, что обеспечивает связь с задачами оптимизации бизнес-процессов организации. Затрагиваются также процессы управления ИТ в организации, что выражается в описании технологической архитектуры и архитектуры приложений [5].

Архитектура предприятия чаще всего включает в себя следующие *компоненты*:

- миссия и стратегия организации;
- ее цели и задачи;
- бизнес-архитектура;
- системная архитектура.

Корпоративные *миссия и стратегия* описывают основные направления будущего развития предприятия и ставят *цели и задачи* на

долгосрочный период. *Бизнес-архитектура*, основываясь на миссии, стратегии развития организации и долгосрочных бизнес-целях, определяет бизнес-процессы, необходимые для выполнения поставленных задач, информационные и материальные потоки, а также описывает организационную структуру компании. *Системная архитектура* – совокупность методологических, технических и технологических решений, помогающих обеспечить информационную поддержку деятельности предприятия. Эта часть архитектуры предприятия определяется *бизнес-архитектурой* компании, и подразделяется на архитектуру приложений, данных и техническую архитектуру [5].

Архитектура приложений включает в себя [7]:

- прикладные системы и программные средства, поддерживающие реализацию бизнес-процессов компании;
- интерфейсы взаимодействия прикладных систем между собой и с внешними системами или теми, кто использует данные;
- средства и методы разработки и сопровождения приложений.

Архитектура данных включает [7]:

- базы данных и хранилища данных;
- различные системы управления базами данных или хранилищами данных;
- правила и средства доступа к данным.

Техническая архитектура состоит из сетевой архитектуры и архитектуры платформ.

Сетевая архитектура состоит из [7]:

- локальных и территориальных вычислительных сетей;
- коммуникационных протоколов, различных сервисов и системы адресации;
- аварийных планов по обеспечению бесперебойной работы сетей в условиях чрезвычайных обстоятельств.

Архитектура платформ включает [7]:

- аппаратные средства вычислительной техники – различные серверы, рабочие станции, накопители и другое компьютерное оборудование;
- операционные и управляющие системы, утилиты и офисные программные системы;
- аварийные планы по обеспечению бесперебойной работы аппаратуры (главным образом – серверов) и баз данных в условиях чрезвычайных обстоятельств [5].

Разработка архитектуры предприятия является процессом, основанным на бизнес-стратегии, который координирует идущие параллельно процессы создания бизнес-архитектуры, архитектуры инфор-

мации, архитектуры прикладных систем и технологической архитектуры. Таким образом, архитектура предприятия является целостным описанием ключевых стратегий организации, связанных с бизнесом, информацией, прикладными системами и технологиями, а также их влиянием на функции и бизнес-процессы организации.

Разработка архитектуры предприятия начинается с *исследования предметной области и построения модели* [5].

При разработке архитектуры предприятия обычно используются два вида инструментов: Enterprise Architecture Tools и Business Process Analyze Tools.

Инструменты *Enterprise Architecture Tools* (ЕАТ) включают средства моделирования бизнес-процессов, их основная задача – предоставить возможность моделирования и проектирования архитектуры предприятия с помощью графического языка. Такие инструменты ориентированы в первую очередь на то, чтобы выявить и зафиксировать связи между бизнес-процессами и другими объектами. Инструменты ЕАТ предоставляет следующие возможности:

- описание в единой модели основных составляющих АП (процессов, структур, целей, информации и др.) и их взаимосвязей и возможность работать с единой базой данных (репозиторием) и хранить информацию о деятельности предприятия «в одном месте»;
- наглядное и удобное для восприятия различными категориями специалистов (руководством и сотрудниками компании, аналитиками и специалистами по ИТ, контролирующими и сертифицирующими органами, партнёрами) представление знаний о компании через автоматическую генерацию отчетов из единой модели в текстовом, табличном или графическом виде;
- анализ моделей (количественный и качественный, статический и динамический);
- поддержание единой системы терминов, понятий и их отношений, которая повышает эффективность внутрифирменных коммуникаций.

Инструменты *Business Process Analyze Tools* (ВРАТ) ориентированы на моделирование и управление бизнес-процессами. Средства ВРАТ позволяют не только моделировать бизнес-процессы, но и проводить мониторинг их количественных параметров, что позволяет выявлять узкие места и оптимизировать бизнес-процессы. Такие инструменты ориентированы на анализ и оптимизацию конкретных бизнес-процессов организации.

Анализ инструментов моделирования

В соответствии с аналитическим обзором Gartner лидерами в категории инструментов управления АП являются MEGA, ARIS (Software AG), Enterprise Studio (BiZZdesign).

Для задач менеджмента особо актуальны инструменты, позволяющие эффективно работать с бизнес-архитектурой как компонентом АП (моделировать, анализировать и проектировать). Для построения моделей могут применяться универсальные графические редакторы, например MS Visio. Однако по мере роста масштаба и сложности задач возникает потребность в специализированных инструментах. Сложные задачи проектирования АП требуют соответствующего комплексного решения, включающего методологию проектирования АП, блок технологической поддержки, анализа, а также деятельность по сопровождению и поддержке [2].

Таким образом, на практике систему управления архитектурой предприятия дополняют системой управления портфелями и проектами (PPM), инструментами финансового планирования и т.д. Также при проектировании АП используются инструменты BPAT. В эту группу включают BPMS- (Business Process Management Suites) или BPM-системы.

Задача описания бизнес-процессов на данный момент является актуальной для многих организаций [3]. Чаще всего процессы уже описаны в той или иной форме, но графическая форма описания явно превалирует над текстовым и табличным форматом. Для ИТ-специалистов описание процессов является отличным способом детализировать требования бизнеса, а также позволяет найти общий язык с ключевыми пользователями. Описание бизнес-процессов строится в целях их оптимизации. Именно тогда эффект от выполненных работ будет максимальным. Описывая «сквозные» бизнес-процессы, «проходящие» через несколько функциональных подразделений, можно обнаружить множество организационных и информационных разрывов между ними, что даёт направления для совершенствования системы [6].

Идея применения специализированного инструментария для моделирования и анализа деятельности в целом и бизнес-процессов в частности, заключается в создании централизованного комплекса взаимосвязанных моделей, описывающих различные предметные области. Они регулярно актуализируются, что даёт возможность их использования сотрудниками различных подразделений. Такая централизация методологии и инструментария позволяет уменьшить затраты на описание деятельности на уровне всей компании. Нужно отметить, что существующий инструментарий даёт возможность не только выпол-

нить описание и регламентацию бизнес-процессов, но и провести их детальный анализ (в частности, средствами имитационного моделирования и анализа стоимости бизнес-процессов). Более продвинутые инструменты позволяют анализировать полноту и непрерывность описанных бизнес-процессов, а также настраивать алгоритмы анализа самостоятельно. Они обеспечивают управление доступом к моделям, формирование регламентов, автоматизацию маршрутов согласования моделей, перенос моделей в среду исполнения [1].

В табл. 1 представлены примеры самых известных инструментальных средств EAT и BPAT.

Таблица 1. Примеры различных видов инструментов для разработки архитектуры предприятия

Название	Задачи		Языки моделирования (нотации)					
	EAT	BPAT	IDEF	BPMN	UML	ARIS	Archi	EPC
AllFusion Process Modeler		+	+					
Archi	+						+	
ARIS Express		+		+	+	+		+
Bizage Modeller		+		+				
BonitaBPM		+		+				
Business Studio		+						
Eclipse	+				+			+
HorusBPM		+		+	+			
Modelio	+			+	+			
RunaWFE		+		+				
MS Visio	+							+

В табл. 1 приведено краткое описание самых популярных продуктов, которые используются при разработке архитектуры предприятия.

Помимо систем, показанных в таблице, существует множество российских и зарубежных инструментов, обладающих функциональностью описания бизнес-процессов и использующих методологии IDEF, BPMN или EPC. Очень часто разработчики систем BPMS распространяют инструменты описания процессов бесплатно, чтобы облегчить последующий перенос моделей процессов в среду исполнения, однако функциональность этих решений часто недостаточна для решения серьёзных задач.

Если анализировать перспективы рынка инструментариев BPAT,

то можно ожидать, что число лёгких и бесплатных инструментов для описания бизнес-процессов будет только расти. При этом преобладать будут методологии BPMN и EPC, тогда как IDEF будет уходить на задний план. В то же время полнофункциональные инструменты для описания бизнес-процессов в крупных компаниях будут и дальше наращивать свою функциональность в части средств анализа бизнес-процессов и соответствующих показателей, а также интеграции с системами автоматизации.

Структура системы автоматизации проектирования архитектуры предприятия

При проектировании архитектуры предприятия используемые для решения различных задач инструменты, позволяющие моделировать архитектуру и анализировать бизнес-процессы, не взаимодействуют между собой. Отсутствует преемственность в решении задач, требующая интеграции описанных средств, гибкой настройки на решаемые задачи.

Предлагается подход, основной целью реализации которого является интеграция Enterprise Architecture Tools и Business Process Analyze Tools через создание платформы интеграции, обеспечивающей гибкое конфигурирование средств моделирования и анализа.

Оба вида инструментов EAT и BPAT, необходимых для разработки архитектуры предприятия, используют формальные модели и графические нотации. Создание средств управления визуальными (графическими) моделями, таким образом, становится основой для создания платформы интеграции инструментов проектирования АП. Основная функция системы – автоматизация трансформации моделей при переходе от задачи моделирования взаимосвязей компонентов архитектуры предприятия между собой к задаче анализа бизнес-процессов, реализующихся с использованием компонентов бизнес-архитектуры, архитектуры данных, приложений и технологической инфраструктуры [8, 9].

В результате анализа популярных средств проектирования архитектуры предприятия (табл. 1), языков, используемых в рассматриваемых инструментах (нотаций), были определены основные требования к платформе интеграции, разработана общая архитектура средств автоматизации проектирования архитектуры предприятия. Основные компоненты системы – средства, реализующие базовую функциональность языковых инструментариев (DSM-платформ), на основе которых решаются задачи автоматизации построения и трансформации моделей при переходе от одной задачи проектирования к другой (рис. 1).

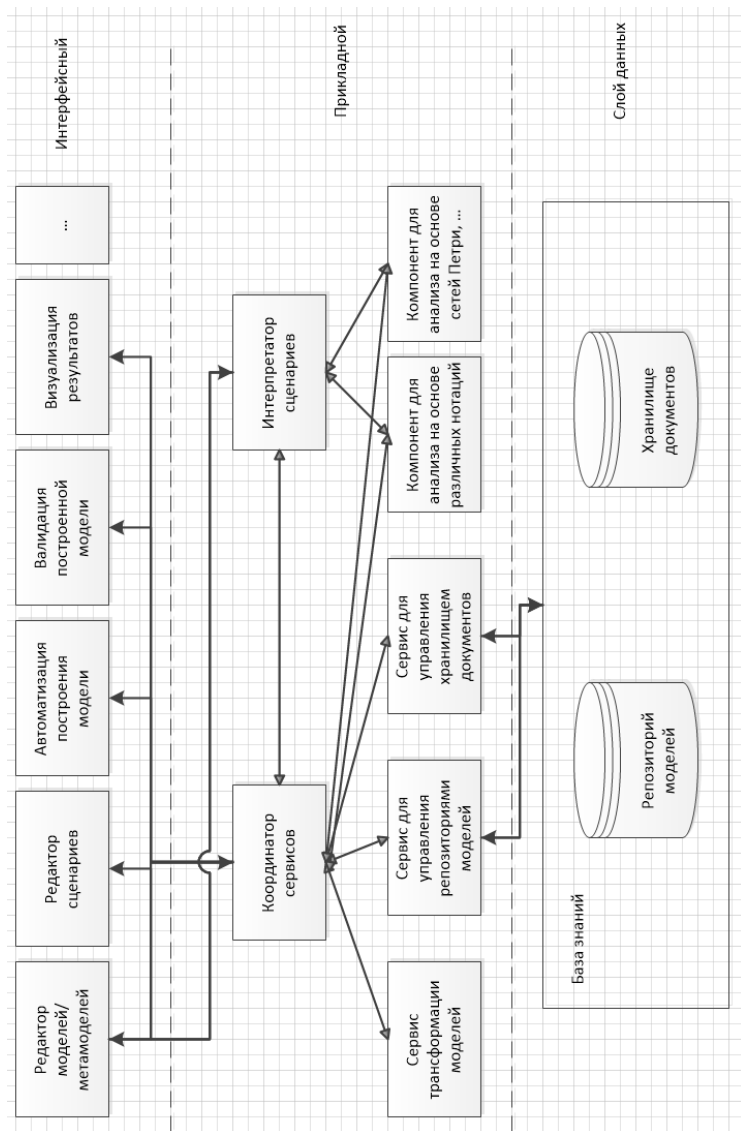


Рис. 1. Архитектура системы автоматизации проектирования архитектуры предприятия

Система должна давать возможность пользователю

- создавать и редактировать модели и метамодели (языки), задавать правила трансформаций моделей;
- создавать сценарии выполнения задач проектирования с использованием различных инструментальных средств;
- автоматизировать построение моделей на основе накопленных знаний и анализа информации, получаемой из различных источников;
- выполнять валидацию построенных моделей;
- выполнять анализ и визуализировать полученные результаты.

На прикладном уровне были выделены интерпретатор сценариев и координатор сервисов, позволяющий сервисам взаимодействовать и реализовывать необходимые функции системы. В слое данных были выделены репозиторий моделей и хранилище документов, позволяющих накапливать знания и опыт экспертов для последующего использования при автоматическом построении аналогичных моделей.

Заключение

Анализ средств проектирования архитектуры предприятия показал, что в данной области существуют некоторые проблемы интеграции различных видов инструментов между собой, что значительно затрудняет процесс моделирования и проектирования архитектуры предприятия. В результате анализа популярных инструментов и нотаций были выделены основные функции системы автоматизации проектирования архитектуры предприятия и выполнено эскизное проектирование архитектуры системы, реализация которой должна обеспечить снижение трудоёмкости разработки и анализа моделей.

Библиографический список

1. *Волкова В. Н.* Сопоставление определений систем и подходов к их исследованию и проектированию / *В.Н. Волкова* // Проблемы управления в социальных системах. – 2012. – 4(6). С.36-50.
2. *Вотинцева В.О.* Применение метода анализа иерархии для выбора методик проектирования системной архитектуры предприятия / *В.О. Вотинцева, А.И. Дерябин, М.С. Сахипова* // Математика и междисциплинарные исследования. – 2016: сб. док. всеросс. науч.-практ. конф. молодых ученых с междунар. участием (г. Пермь, 16-19 мая 2016 г.). – Пермь: Издательский центр ПГНИУ, 2016. – С. 201-208.

3. *Гаврилова Т. А.* Онтологический подход к управлению знаниями при разработке корпоративных информационных систем / *Т.А. Гаврилова* // *Новости искусственного интеллекта.* – 2003. – № 2. – С. 24-30.
4. *Зараменских Е.П.* Основы бизнес-информатики / *Е.П. Зараменских.* – М.: Юрайт, 2005.
5. *Зараменских Е.П.* Архитектура предприятия : учебник для бакалавриата и магистратуры / *Е.П. Зараменских, Д.В. Кудрявцев, М. Ю. Арзуманян.* – М. : Юрайт, 2018.
6. *Барановская Т.П.* Архитектура компьютерных систем и сетей / *Т.П. Барановская.* – М.: Финансы и статистика, 2003.
7. *Галактионов В.И.* Системная архитектура и ее место в архитектуре предприятия [Электронный ресурс] / *В.И. Галактионов* // *Директор информационной службы.* – 2002. – № 5. – Режим доступа: <http://www.ospr.ru/cio/2002/05/172142>.
8. *Котелов А.* Краткий обзор продукта Archi – ArchiMate [Электронный ресурс] / *А. Котелов.* – Холдинг РБК. – 2012. – Режим доступа: http://club.cnews.ru/blogs/entry/kratkij_obzor_produkta_archi.
9. *Точилкина Т.Е.* Моделирование архитектуры предприятия с Archi [Электронный ресурс] / *Т.Е. Точилкина* // *Электронный научно-практический журнал «Экономика и менеджмент информационных технологий».* – 2014. – № 11. Режим доступа: <http://ekonomika.snauka.ru/2014/11/6308>.

А.О. Сухов¹, Л.Н. Лядова¹, С.А. Порязов²,

¹Национальный исследовательский университет
«Высшая школа экономики»
(Пермский филиал)
ASuhov@hse.ru, LLyadova@hse.ru

²Институт математики и информатики БАН
(София, Болгария)
stoyan@cc.bas.bg

ГИПЕРГРАФЫ С ПОЛЮСАМИ КАК ОСНОВА СОЗДАНИЯ РЕДАКТОРОВ ВИЗУАЛЬНЫХ ЯЗЫКОВ

В статье рассматриваются различные способы формализации описания визуальных моделей. Предлагается новая модель для создания графических языков – гиперграф с полюсами, обеспечивающая возможность определения и реализации не только новых визуальных языков, но и дающая основу для реализации операций над моделями, построенными с помощью этих языков. Предложенная модель – расширение понятия графа с полюсами, учитывающая специфику реализации графических редакторов для DSM-платформ, реализующих не только средства определения языков и создания моделей, но и возможности задания трансформаций моделей, построенных с их помощью, с одного языка на другой.

Ключевые слова: визуальное моделирование, графические модели, мета-графы, гиперграфы, графы с полюсами, редакторы моделей, предметно-ориентированное моделирование, языковой инструментарий.

Введение

Существует множество решений для работы с графическими моделями: различные CASE-системы и системы имитационного моделирования, системы автоматизации проектирования и управления технологическими процессами включают средства создания и анализа моделей. Средства визуального моделирования, основанные на применении специализированных предметно-ориентированных языков, упрощают создание моделей, служат основой для решения задач ана-

лиза. Однако разработка этих средств связана с определёнными проблемами, является трудоёмкой задачей. Решение этой проблемы – применение при разработке новых языков и систем моделирования специального программного обеспечения – языковых инструментариев (или DSM-платформ). Усилия многих разработчиков направлены на создание языковых инструментариев, их совершенствование [1, 2, 4, 5, 6, 14]. Такая активность характеризует направленный переход от использования стандартизованных графических нотаций к созданию собственных языков моделирования, на основе которых создаются предметно-ориентированные визуальные модели, отражающие особенности конкретных областей, специфику решаемых задач.

Возможности существующих систем визуального моделирования, создания и анализа моделей, во многом определяются математическим аппаратом, на базе которого разработаны редакторы моделей [7, 8, 9, 10, 13].

Для построения визуальных языков в DSM-платформах в качестве математической модели используются различные типы графов. В [6] показано, что использование ориентированного псевдо-метаграфа даёт значительные преимущества по сравнению с другими предлагаемыми типами графов (гиперграфами, *hi*-графами и др.). Однако существует более мощная формальная модель, которая обеспечивает определение на её основе всех рассматриваемых типов графов, – графы с полюсами, или *P*-графы [3, 11].

В [7] были сформулированы общие требования к редакторам визуальных моделей для DSM-платформ, предложена объектная модель для реализации редактора, основанная на *P*-графах. Предложенный подход обеспечивает реализацию всех перечисленных требований, но опыт разработки показал, что усовершенствование выбранной математической модели может обеспечить большую выразительность создаваемых моделей, решить проблему при определении различных типов трансформаций, задаваемых для выполнения переходов от использования одной графической нотации к другой, для решения задачи детализации моделей, их декомпозиции.

Задача данного исследования – создание новой формальной модели, обеспечивающей выполнение различных операций над моделями, создаваемыми с использованием предметно-ориентированных языков (Domain Specific Languages, DSL), обобщающей преимущества различных графовых представлений при решении задач построения и анализа моделей в различных областях исследований.

Определение гиперграфа с полюсами

На основе анализа графических редакторов, используемых графических нотаций и языков визуального моделирования выделены следующие требования к представлению и построению моделей:

– Должна быть обеспечена возможность создания иерархических структур: модель может содержать элементы (вершины), каждый из которых имеет сложную организацию, требующую детализации (расшифровки, декомпозиции).

– Элементы модели могут быть связаны друг с другом, причём создаваемые в модели связи могут отражать сложные взаимоотношения, связи между моделируемыми объектами, а детализация моделей может потребовать не только детализировать организацию элементов-вершин в моделях, но и детализировать связи между ними.

Эти требования могут быть решены при использовании различных типов графов в качестве формальной основы при создании редакторов и средств анализа моделей, но каждый из вариантов обладает определёнными ограничениями, что приводит к усложнению модели, затрудняет её восприятие, обработку, анализ.

Обобщим различные математические модели, используемые при разработке средств визуального моделирования и анализа графовых моделей, обладающие максимальными выразительными возможностями. В качестве основы рассмотрим три типа графов.

Определение 1. *Метаграф* – это упорядоченная пара $G = (X, E)$, где $X = \{x_i\}, i = \overline{1, n}$ – конечное непустое множество вершин, E – множество рёбер графа. Каждое ребро $e_k = (V_i, W_i)$, $k = \overline{1, m}$, $V_i, W_i \subseteq X$, причём $V_i \cup W_i \neq \emptyset$, т.е. каждое ребро метаграфа соединяет два подмножества множества вершин.

Если ребро метаграфа является направленным (дугой), то граф называется ориентированным метаграфом.

Метаграфы используются в системах, применяемых для поиска и визуализации связей, зависимостей между объектами в различных предметных областях (например, в системах Data Mining, которые занимаются обнаружением ассоциаций и последовательностей, и т.п.). Кроме этого, метаграфы применяются в системах, в которых все объекты различной природы можно разделить на несколько групп, между объектами которых устанавливаются связи (в системах моделирования, например, это могут быть множество сущностей, атрибутов, операций над ними и т.п.).

Определение 2. *Гиперграф* представляет собой пару (X, E) , где X – непустое множество объектов некоторой природы, называемых *вершинами гиперграфа*, а E – семейство непустых подмножеств множества X , называемых *гиперрёбрами*.

Таким образом, гиперребро объединяет вершины, представляющие некоторые объекты, характеристики и т.п., фиксируя связи различной природы между ними. Однако такое описание не даёт чёткого представления о структуре и природе связей, семантике зависимостей, поэтому не каждая предметная область или задача может быть описана с помощью гиперграфа.

Определение 3. *Граф с полюсами* – это упорядоченная тройка $G = (P, V, W)$, где $P = \{p_1, \dots, p_n\}$ – абстрактное множество – множество внешних полюсов графа, $V = \{v_1, \dots, v_n\}$ – непустое множество вершин, $W = \{w_1, \dots, w_n\}$ – множество связей; при этом:

1. Каждая вершина $v \in V$ – это подмножество множества полюсов P ($v = \{p_{v_1}, \dots, p_{v_n}\}$) и $\forall v_i \forall v_j \in V [i \neq j \rightarrow v_i \cap v_j = \emptyset]$, т.е. V – множество взаимно не пересекающихся подмножеств полюсов.

2. В каждой вершине $v \in V$ выделяются два подмножества: $I(v)$ и $O(v)$ входных и выходных полюсов: $\forall v \in V \exists I(v) \subset P, \exists O(v) \subset P [I(v) \cup O(v) = v]$, причём эти множества могут пересекаться, если эти множества не заданы, будем считать, что $I(v) = O(v) = v$.

3. Множество рёбер, описывающих связи между вершинами, определяется как подмножество множества всех пар полюсов, причём $W \subset P \times P \setminus \text{diag}(P \times P), \forall v \in V \forall p \in v \forall r \in v [(p, r), (r, p) \notin W]$, т.е. полюс не может быть соединён сам с собой, и связи не могут быть установлены между полюсами, принадлежащими одной вершине.

Для графа с полюсами определены операции добавления/удаления полюсов, вершин, рёбер, а также операции расшифровки вершин графами, что позволяет строить иерархические модели сложных систем и процессов, выполняя их пошаговую детализацию. Эти операции могут быть использованы в качестве основы как для разработки графических моделей, так и для их преобразований (трансформаций) на основе графовых грамматик. Однако ограничение на создание связей между полюсами одной и той же вершины затрудняет реализацию редактора моделей, где допускаются такие связи, что не является исключением при реализации языковых инструментариев для создания визуальных DSL.

Обобщение приведённых формальных моделей привело к определению нового типа графов – *гиперграфа с полюсами (HP-граф)*.

Определение 4. *Гиперграф с полюсами* – это упорядоченная тройка $G = (P, V, W)$, где $P = \{p_1, \dots, p_n\}$ – абстрактное множество – множество внешних полюсов графа, $V = \{v_1, \dots, v_m\}$ – непустое множество вершин, $W = \{w_1, \dots, w_l\}$ – множество связей; при этом:

1. Каждая вершина $v \in V$ – это подмножество множества полюсов P ($v = \{p_{v_1}, \dots, p_{n_v}\}$) и $\forall v_i \forall v_j \in V [i \neq j \rightarrow v_i \cap v_j = \emptyset]$, т.е. V – множество взаимно не пересекающихся подмножеств полюсов.

2. В каждой вершине $v \in V$ выделяются два подмножества: $I(v)$ и $O(v)$ входных и выходных полюсов: $\forall v \in V \exists I(v) \subset P, \exists O(v) \subset P [I(v) \cup O(v) = v]$, причём эти множества могут пересекаться; если эти множества не заданы, будем считать, что $I(v) = O(v) = v$, т.е. по умолчанию, если множества полюсов вершины не определены, вершина представляется одним полюсом.

3. Множество рёбер, описывающих связи между вершинами, определяется как подмножество множества всех комбинаций (подмножеств) полюсов, т.е. ребро $w \in W$ – это множество полюсов: $w = P_w = \{p_{k_1}, \dots, p_{k_w}\} \subset P$, где полюса $p \in P_w$ могут принадлежать одной или нескольким вершинам, между которыми есть связи, и $P_w \neq \emptyset$.

Отличие от приведённого определения графа с полюсами – возможность создания «гиперрёбер», которые объединяют не отдельные полюсы из множества полюсов, принадлежащих различным вершинам, а множества полюсов, которые могут принадлежать одной или нескольким вершинам. Это позволяет визуализировать сложные связи как между парами вершин, так и между множествами вершин, причём вершина может быть связана сама с собой. В отличие от гиперграфа связи могут представлять более детальную информацию, более чётко отражающую структуру моделируемых систем, связи между объектами, их направленность.

При необходимости к «гиперребру» в *HP*-графе, как и к вершине, можно применить операцию расшифровки, раскрывающую структуру связи при декомпозиции модели или при её трансформации при переходе от использования одной графической нотации к другой. При выполнении этой операции связь детализируется: либо выделяются пары полюсов, входящие в соответствующее «гиперребро», либо определяется структура в виде нового *HP*-графа, расшифровывающая связь через новые вершины, появляющиеся на более глубоком уровне детализации модели. Направление связей при расшифровке определяется типами полюсов – их принадлежностью множествам входных и выходных полюсов соответствующих вершин.

Упрощённый пример *HP*-графа показан на рис. 1. Здесь множество вершин представляет устройства. Каждое устройство имеет множества полюсов – портов. Устройства связаны через шину, которая представляется «гиперребром», объединяющим полюса вершин, представляющих устройства компьютера, обменивающиеся информацией через шину. Причём на данном уровне детализации нет никакого представления об организации этих устройств и о структуре шины.

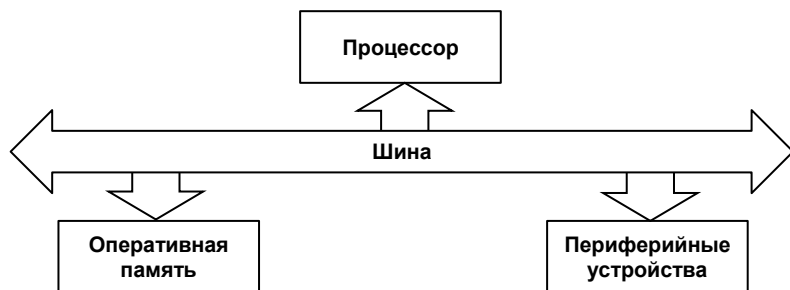


Рис. 1. HP-граф – пример модели

При детализации моделей каждая вершина может быть расшифрована графом, отображающим структуру соответствующего устройства (например, функциональные блоки процессора и связи между ними). Расшифровано может быть и «гиперребро», например: «Шина» может быть декомпозирована на шины адреса и данных, могут быть выделены также линии управления. Каждая из выделенных при декомпозиции «гиперребра» линий будет представлена парой полюсов (портов), которые соединяются через неё.

Заключение

При сравнении различных формализмов, используемых при разработке графических моделей, показано [6], что наиболее подходящим формализмом описания метаязыка при создании DSM-платформы являются графовые грамматики, построенные на основе псевдо-метаграфов. Этот вид графов позволяет структурировать модель за счет объединения вершин в множества. Предлагаемая модель позволяет объединять в множества и рёбра. Это позволяет при описании сложных моделей большой размерности сделать их менее громоздкими, более простыми для восприятия. При этом операции расшифровки вершин и «гиперрёбер» в *HP*-графе позволяют выполнять декомпози-

цию моделей при их детализации и анализе, при определении правил трансформации моделей для решения различных задач.

Подобные средства построения моделей могут быть полезны при разработке моделей бизнес-процессов и их пошаговой детализации [6, 7], при разработке моделей телекоммуникационных систем и их преобразованиях в ходе анализа, оценки и оптимизации [12], т.е. при решении задач создания и анализа сложных моделей большой размерности для различных областей. Все эти возможности должны быть учтены при разработке языковых инструментариев, DSM-платформ, что обеспечит их максимальную гибкость, масштабируемость при разработке визуальных языков и моделей.

Применение *HP*-графа даёт возможность реализовать новые характеристики и операции при разработке объектной модели для графического редактора [7] и метаязыка для DSM-платформы.

Библиографический список

1. *Брыксин Т.А.* Платформа для создания специализированных визуальных сред разработки программного обеспечения / *Т.А. Брыксин, А.Н. Терехов* // СПб.: Санкт-Петербургский государственный университет, 2016.
2. *Лядова Л.Н.* Языковой инструментарий системы MetaLanguage / *Л.Н. Лядова, А.О. Сухов* // Математика программных систем: межвуз. сб. науч. ст. Пермь: Изд-во Перм. гос. ун-та, 2008. – Вып. 5. – С. 40-51.
3. *Миков А.И.* Автоматизация синтеза микропроцессорных управляющих систем / *А.И. Миков* // Иркутск: Изд-во Иркут. Ун-та, 1987.
4. *Сухов А.О.* Среда разработки визуальных предметно-ориентированных языков моделирования / *А.О. Сухов* // Математика программных систем: межвуз. сб. науч. ст. Пермь: Изд-во Перм. гос. ун-та, 2008. – Вып. 5. – С. 84-94.
5. *Сухов А.О.* Теоретические основы разработки DSL-инструментария с использованием графовых грамматик / *А.О. Сухов* // Информатизация и связь. – 2011. – № 3. – С. 35-37.
6. *Сухов А.О.* Разработка инструментальных средств создания визуальных предметно-ориентированных языков : дис. ... канд. физ.-мат. наук: 05.13.11 / *Сухов Александр Олегович.* М., 2013. 256 с.
7. *Филатов Д.Ю.* Разработка редактора визуальных моделей, основанного на Р-графах / *Д.Ю. Филатов, Л.Н. Лядова* // В кн.: Технологии разработки информационных систем (ТРИС-2017):

- Материалы VIII Международной научно-технической конференции. Ростов н/Д : Южный федеральный университет, 2017. С. 113-118.
8. *Dietrich H.-A.* Graphical Model Editors for Meta-Modelling Tools – Requirements, Conceptualisation, and Implementation / *H.-A. Dietrich, D. Breuker, M. Steinhorst, P. Delfmann, J. Becker* // Enterprise Modelling and Information Systems Architectures. – 2013. – Vol. 8, No. 2. – P. 42-78.
 9. *Goik D.* Hypergraph grammar based adaptive linear computational cost projection solvers for two and three dimensional modeling of brain / *D. Goik, M. Sieniek, M. Wozniak, A. Paszynska, M. Paszynski* // Procedia Computer Science. – 2014. Vol. 29. – P. 1002–1013.
 10. *Maximova M.* Analysis of Hypergraph Transformation Systems in AGG based on \mathcal{M} -Functors: Extended Version / *M. Maximova, H. Ehrig, C. Ermel* // Electronic Communications of the EASST. – 2013. Vol. 58: Graph Transformation and Visual Modeling Techniques. – DOI: <http://dx.doi.org/10.14279/tuj.eceasst.58.841.836>.
 11. *Mikov A.I.* Program Tools and Language for Network Simulation and Analysis / *A.I. Mikov, E.B. Zamiatina* // In: Proceedings of International Science and Technology Conference (Modern Networking Technologies) (MoNeTeC). Moscow, 2014. P. 1-9. DOI Bookmark: <http://doi.ieeecomputersociety.org/10.1109/MoNeTeC.2014.6995591>.
 12. *Poryazov S.* Conceptual and Analytical Models for Predicting the Quality of Service of Overall Telecommunication Systems / *S. Poryazov, E. Saranova, I. Ganchev* // Autonomous Control for a Reliable Internet of Services. Lecture Notes in Computer Science, vol. 10768. Springer, Cham, 2018. P. 151-181.
 13. *Rossini A.* A Graph Transformation-Based Semantics for Deep Metamodeling / *A. Rossini, J. de Lara, E. Guerra, A. Rutle, Y. Lamo* // In: AGTIVE 2011: 4th International Symposium on Applications of Graph Transformations with Industrial Relevance. Budapest, Hungary, 2011. Vol. LNCS, 7233. P. 19-34.
 14. *Tolvanen J.-P.* Model-Driven Development Challenges and Solutions: Experiences with Domain-Specific Modelling in Industry / *J.-P. Tolvanen, S. Kelly* // Proceedings of MODELSWARD 2016, 4th International Conference on Model-Driven Engineering and Software Development. – Rome: SCITEPRESS. 2016.

ABSTRACTS

Gorodilov Aleksej Yurievich, Makurin Roman Anatolyevich

THE USE OF GENETIC ALGORITHMS IN THE AUTOMATIC SCHOOL TIMETABLE COMPILATION

The article describes the approach to the school timetable compilation based on genetic algorithms. Schedule requirements are divided into two categories: critical, the failure of which leads to the appearance of an incorrect, impracticable schedule, and user, allowing to consider the individual preferences of teachers and features of a school. Critical schedule requirements are described by graph theory language, which allows the use of standard genetic operators without causing many non-viable individuals. User requirements are described in the form of a flexibly adjustable fitness function, which makes it possible to quickly adapt the proposed algorithm to the conditions of a particular educational institution. The algorithm is implemented as a program for the school timetable compilation. The article presents the real data testing results, shows the correctness and advantages of the proposed approach, and marks the development trends.

Keywords: genetic algorithms, timetable compilation, graphs, fitness function.

Dadenkova Alyona Pavlovna

INVESTIGATION OF CHARACTERISTICS OF THE DHT SEARCH SERVICE PROTOCOL MODEL

In this article, an analytical study of the decentralized file-sharing network with the DHT protocol and the conclusion of recommendations on the choice of parameters. The work discusses the general requirements, the proposed fragments of the implementation of its model in the simulation environment AnyLogic. The results of experiments on the investigation of the protocol and the performance analysis of the data retrieval of the file-exchange network are presented and presented.

Keywords: simulation modeling, data search protocols, decentralized file-sharing networks, AnyLogic, DHT, P2P.

Zamyatina Elena Borisovna, Kudryavcev Andrei Vladislavovich

SIMULATION OF ROUTING ALGORITHMS FOR AD HOC NETWORKS USING AGENT TECHNOLOGIES

This paper discusses various classes of ad hoc networks, routing algorithms, and more precisely the routing algorithms of FANET networks as well as simulation tools TriadNS which may be used for ad hoc network simulation model investigations. The feasibility of using the agent paradigm of simulation modeling is considered and an attempt is made to develop agents in TriadNS.

Keywords: simulation, agent technology, routing protocols, ad-hoc networks.

Kon Efim Lvovich, Dadenkov Sergey Aleksandrovich

ANALYSIS OF PROTOCOL STACKS OF FIELDBUS NETWORKS WITH RANDOM MULTIPLICATIVE CSMA ACCESS

An analysis is performed of common communication protocol stacks of sensory fieldbus networks with random multiple access algorithms for CSMA nodes to a shared transmission medium. A correlation of network protocol stacks is carried out with a seven-level open systems interconnection model. A brief description of the access algorithms used in the analyzed networks and their differences are presented. As a result of the analysis, the seven-level communication stack of the LonWorks network protocols, which is characterized by a parametric and functional commonality with all of the analyzed stacks, is highlighted. The result obtained allows us to perform a further generalized study of networks, the results of which can be used for a wide range of sensory fieldbus networks with random access.

Keywords: communication protocol stack, industrial bus, sensor network, CSMA, fieldbus, network, LonWorks.

Makarov Vladislav Vladimirovich, Lanin Viacheslav Vladimirovich

GATE DEVELOPER PROFILING FOR IDENTIFICATION OF MEMORY OVERFLOW REASONS

Paper is prepared at a research on project No. 17-05-0020 supported by the Scientific Fund of the National Research University Higher School of Economics, with in the state support of leading universities of the Russian Federation "5-100" (Program in 2017-2018). In the article the problem of overflow of memory in the application for linguistic analysis of GATE Developer is investigated, the reasons of emergence of a problem are listed, suitable tools for carrying out profiling are selected, profiling by JProfiler with detection of patterns is carried out and options of solutions are given.

Keywords: profiling, memory overflow, linguistic analysis.

Markvirer Vladlena Dmitrievna, Suvorov Alexandr Olegovich

PATTERNS, TECHNOLOGIES AND TOOLS FOR SPORT SOFTWARE MOBILE APPLICATIONS DEVELOPMENT

This paper represents analysis of software development patterns, tools and development environments for creating mobile applications, and realization of analytic technologies for data stored in application. The practical part of the work is a brief description of creation cyclist mobile application by the selected methods, confirming the possibility of implementing the tasks set by the customer using the described technologies.

Keywords: MVC, Android, Android Studio, Java, mobile application, sport.

Mikov Alexander Ivanovich

DYNAMIC GEOMETRIC GRAPHS IN MULTI-CONNECTED DOMAINS

Mathematical models of mobile wireless computer networks in the form of geometric graphs are investigated. The focus is on the network location model. The influence of the parameters of a doubly connected domain on the probability characteristics of the connectivity of random geometric graphs is studied in detail. A model of a multiply connected region and the influence of its shape factor on the possibility of forming routes are proposed and studied. An example of changing a dynamic graph in a multiply connected domain is considered. The main provisions are illustrated with graphs made on the basis of statistical analysis.

Keywords: random graph, connectivity, ad hoc network, route, dynamic graph, moving model.

Plaksin Mikhail Aleksandrivich

A METHOD OF MANAGEMENT QUALITY ASSESSMENT FOR ORGANIZATION SYSTEM

The article describes the methodic of applying the Anticipatory Failure Determination method to assess the quality of management of an organizational system. To identify potential shortcomings and assess their significance, collective anonymous examination and statistical processing of expert assessments are used. As a result, ATD was useful for assessing the quality of the management process and identifying ways to improve it. As an example, the application of the described methodology to the assessment of the quality of the educational process at the Department of Business Informatics of the HSE-Perm is considered. But there are no obstacles to the use of the proposed method in other cases.

Keywords: quality of management, diversionary analysis method, collective anonymous examination, statistical analysis.

Sakhpova Marina Stanislavovna, Lyadova Lyudmila Nickolaevna

ON APPROACH TO SOLUTION OF PROBLEM OF ENTERPRISE ARCHITECTURE DESIGN AUTOMATION

The tools used at design of enterprise architecture are analyzed, their characteristics are described in article . The output that the main problem complicating design process, leading to increase in labor input of work of systems analysts and engineers is the lack of integration tools of the used tools is proved. Approach to a solution of an objective is offered and the general architecture of the developed software is described.

Keywords: enterprise architecture, design tools, design automation system, system architecture.

*Sukhov Alexander Olegovich, Lyadova Lyudmila Nickolaevna,
Poryazov Stoyan Atanasov*

ON APPROACH TO SOLUTION OF PROBLEM OF ENTERPRISE ARCHITECTURE DESIGN AUTOMATION

Different ways of the visual model descriptions formalization are considered. The hypergraph with poles is offered as a new formal model for graphic languages creation. This model provides possibility of definition and implementation of new visual languages and it gives a basis for realization of operations over the models constructed with these languages. The suggested model is an extension of the graph with poles concept for considering specifics of graphics editors implementation for the DSM platforms. The language toolkits include means of languages definition and tools for models creation, but components for model transformations are the most important in modern DSM platform. New formal model provides flexible tools for decomposition and detailing of models and language transformations feasibility.

Keywords: visual modeling, graphic models, meta-graphs, hypergraphs, graph with poles, model editors, domain specific modeling, language toolkits.

СВЕДЕНИЯ ОБ АВТОРАХ

Городилов Алексей Юрьевич – кандидат технических наук, доцент кафедры математического обеспечения вычислительных систем Пермского государственного национального исследовательского университета (стр. 3).

Даденков Сергей Александрович – кандидат технических наук, доцент кафедры автоматики и телемеханики Пермского национального исследовательского политехнического университета (стр. 28).

Даденкова Алена Павловна – студент магистратуры, кафедра автоматики и телемеханики Пермского национального исследовательского политехнического университета (стр. 12).

Замятина Елена Борисовна – кандидат физико-математических наук, доцент, доцент кафедры информационных технологий в бизнесе Национального исследовательского университета «Высшая школа экономики» (Пермский филиал); доцент кафедры математического обеспечения вычислительных систем Пермского государственного национального исследовательского университета (стр. 18).

Кон Ефим Львович – кандидат технических наук, профессор кафедры автоматики и телемеханики Пермского национального исследовательского политехнического университета (стр. 28).

Кудрявцев Андрей Владиславович – студент магистратуры, кафедра математики и компьютерных наук Уральского федерального университета (стр. 18).

Ланин Вячеслав Владимирович – старший преподаватель кафедры информационных технологий в бизнесе Национального исследовательского университета «Высшая школа экономики» (Пермский филиал) (стр. 44).

Лядова Людмила Николаевна – кандидат физико-математических наук, доцент; доцент кафедры информационных технологий в бизнесе Национального исследовательского университета «Высшая школа экономики» (Пермский филиал); доцент кафедры математического обеспечения вычислительных систем Пермского государственного национального исследовательского университета (стр. 87, 97).

Макаров Владислав Владимирович – студент бакалавриата, кафедра информационных технологий в бизнесе Национального исследовательского университета «Высшая школа экономики» (Пермский филиал) (стр. 44).

Макурин Роман Анатольевич – студент магистратуры, кафедра математического обеспечения вычислительных систем Пермского государственного национального исследовательского университета (стр. 3).

Марквирер Владлена Дмитриевна – преподаватель кафедры информационных технологий в бизнесе Национального исследовательского университета «Высшая школа экономики» (Пермский филиал) (стр. 50).

Миков Александр Иванович – доктор физико-математических наук, профессор, зав. кафедрой вычислительных технологий Кубанского государственного университета (стр. 63).

Плаксин Михаил Александрович – кандидат физико-математических наук, доцент; доцент кафедры информационных технологий в бизнесе Национального исследовательского университета «Высшая школа экономики» (Пермский филиал); доцент кафедры математического обеспечения вычислительных систем Пермского государственного национального исследовательского университета (стр. 74).

Порязов Стоян Атанасович – доктор наук, старший научный сотрудник, Институт математики и информатики БАН, руководитель департамента информационного моделирования (стр. 97).

Сахипова Марина Станиславовна – аспирант, кафедра математического обеспечения вычислительных систем Пермского государственного национального исследовательского университета; преподаватель кафедры информационных технологий в бизнесе Национального исследовательского университета «Высшая школа экономики» (Пермский филиал) (стр. 87).

Суворов Александр Олегович – кандидат технических наук, доцент, доцент кафедры информационных технологий и автоматизированных систем Пермского национального исследовательского политехнического университета (стр. 50).

Сухов Александр Олегович – кандидат физико-математических наук, доцент кафедры информационных технологий в бизнесе Национального исследовательского университета «Высшая школа экономики» (Пермский филиал) (стр. 97).

СОДЕРЖАНИЕ

<i>Городилов А.Ю., Макурин Р.А.</i> Применение генетических алгоритмов в системе автоматического составления учебных расписаний	3
<i>Даденкова А.П.</i> Исследование характеристик модели протокола поискового сервиса DHT.....	12
<i>Замятина Е.Б., Кудрявцев А.В.</i> Моделирование алгоритмов маршрутизации ad-hoc-сетей с использованием агентных технологий.....	18
<i>Кон Е.Л., Даденков С.А.</i> Анализ стеков протоколов FIELDBUS-сетей со случайным множественным доступом CSMA	28
<i>Макаров В.В., Ланин В.В.</i> Профилирование GATE Developer для выявления причины переполнения памяти	44
<i>Марквирер В.Д., Суворов А.О.</i> Паттерны, технологии и инструменты мобильной разработки прикладных программ для спорта	50
<i>Миков А.И.</i> Динамические геометрические графы в многосвязных областях	63
<i>Плаксин М.А.</i> Об одном методе оценки качества управления организационной системой.....	74
<i>Сахипова М.С., Лядова Л.Н.</i> О подходе к решению задачи автоматизации проектирования архитектуры предприятия	87
<i>Сухов А.О., Лядова Л.Н., Порязов С.А.</i> Гиперграфы с полюсами как основа создания редакторов визуальных языков.....	97
Abstracts	105
Сведения об авторах.....	109

Научное издание

Математика программных систем

Межвузовский сборник научных трудов

Выпуск 15

Издаётся в авторской редакции

Компьютерная вёрстка *Л.Н. Лядовой*

Подписано в печать 28.12.2018. Формат 60×84 ¹/₁₆.

Усл. печ. л. 6,23. Тираж 100 экз.

Издательский центр

Пермского государственного
национального исследовательского университета.
614990, г. Пермь, ул. Букирева, 15

Отпечатано в ООО «Учебный центр “Информатика”»

614990, г. Пермь, ул. Букирева, 15